

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

**Modelagem Orientada a Agentes
Aplicada a Ambientes Inteligentes
Distribuídos de Ensino:**

JADE
*Java Agent framework for
Distance learning Environments*

por
RICARDO AZAMBUJA SILVEIRA

Tese de Doutorado

Dr.^a Rosa Maria Vicari
Orientadora.

Porto Alegre, janeiro de 2001.

CIP - Catalogação na Publicação

Silveira, Ricardo Azambuja.

Modelagem Orientada a Agentes Aplicada a Ambientes Inteligentes Distribuídos de Ensino – JADE - *Java Agent framework for Distance learning Environments* / por Ricardo Azambuja Silveira. - Porto Alegre: PPGC da UFRGS, 2000.

126 f.:il.

Tese (doutorado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR – RS, 2000. Orientadora: Vicari, Rosa Maria

1. Informática na Educação. 2. Ensino à distância 3. Inteligência Artificial. 4. Redes de computadores.

I. Vicari, Rosa Maria. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Profa. Wrana Panizzi

Pró-Reitor de Ensino: Prof. José Carlos Ferraz Hennemann

Pró-Reitor de Pesquisa: Prof. Carlos Alexandre Netto

Pró-Reitor Adjunto de Pós Graduação: Prof. Philippe Oliver Alexandre Navaux

Diretor do Instituto de Informática: Prof. Philippe Oliver Alexandre Navaux

Coordenador do PPGC: Prof. Carlos Alberto Heuser

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro.

Agradecimentos

Embora seja impossível citar a todas as pessoas que emprestaram seu apoio ou que, de alguma forma, contribuíram com este trabalho, sem correr o risco de omissão, não se poderia deixar de registrar um agradecimento muito especial àqueles que compartilharam desta jornada.

Em primeiro lugar, agradeço aos meus colegas do Curso de Pós-Graduação em Ciência da Computação da UFRGS, Mestrado e Doutorado, que compartilharam comigo estes anos de trabalho e de agradável convivência. De modo especial, aos meus colegas doutorandos Marcelo Ladeira, Lúcia Giraffa, Jugurta Lisboa e Ricardo Bastos que me acolheram e prestaram inestimável apoio.

Aos colegas Alessandro Boeira, Ana Ascencio, Francine Bica e Adriana Pereira por terem apostado neste projeto e cujos trabalhos de elaboração das dissertações foram de fundamental importância para a execução deste trabalho.

A todos os professores do Curso de Pós-Graduação em Ciência da Computação da UFRGS com quem tive a oportunidade de conviver e a todos os funcionários e colaboradores do Instituto de Informática da UFRGS que não medem esforços para o cumprimento de sua missão.

Aos colegas da CRT Brasil Telecom, e à empresa, pelo apoio.

Ao professor Milton Madeira, a Simone Engler e a todos que trabalharam na sua equipe durante esses anos pela contribuição ao trabalho e, especialmente, pelos constantes momentos de incentivo

A meus pais Ieda e Vilson (in memoriam) e a toda a minha família pelo carinho e compreensão.

A minha esposa Marina Nakayama e a Maurício e Júlia, frutos de nosso amor, pelo apoio em todos esses anos e pelas muitas horas roubadas.

A Fundação Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, CAPES, pelo apoio, através do Programa de Apoio à Pesquisa em Educação a Distância – PAPED.

Finalmente, a minha orientadora, Rosa Vicari, pelo carinho com que me acolheu e pela dedicação em todos esses anos e pelo exemplo que tem sido em todos os sentidos.

“STAT ROSA PRISTINA NOMINE, NOMINA NUDA TENEMUS”

(Humberto Eco)

O nome da Rosa...

Sumário

Agradecimentos.....	3
Lista de Abreviaturas	7
Lista de Figuras	8
Lista de Tabelas.....	10
Resumo.....	11
Abstract.....	12
1 Introdução.....	13
1.1Motivação.....	14
1.2 Objetivos, premissas e hipóteses.....	18
1.2.1Objetivo geral.....	18
1.2.2Objetivos específicos.....	19
1.2.3Premissas e hipóteses sob o ponto de vista computacional.....	19
1.2.4Premissas e hipóteses sob o ponto de vista educacional.....	20
2Estado da arte e trabalhos correlatos.....	23
2.1Estado da arte em Educação a Distância.....	23
2.2Estado da arte em IA.....	25
2.2.1Coordenação e comunicação em sistemas multiagentes.....	31
2.2.2Agentes pedagógicos.....	38
2.3Projeto e implementação de Sistemas Multiagentes.....	42
2.4Trabalhos correlatos.....	45
3 O JADE.....	55
3.1Metodologia de análise e modelagem.....	55
3.1.1Análise orientada a agentes.....	56
3.2Princípios pedagógicos.....	59
3.3O modelo organizacional.....	66
3.4O modelo de cooperação.....	70
3.4.1Comunicação.....	70
3.5Modelo de agente.....	81
3.6Implementação.....	84
3.6.1A arquitetura global do JADE.....	86
3.7Desenvolvendo aplicações com o JADE.....	88
4O protótipo Eletrotutor.....	90
4.1O Modelo Organizacional do Eletrotutor III.....	92
4.2O modelo interno dos Agentes.....	93
4.3Estratégia de Ensino.....	94
4.4Modelo de cooperação entre os agentes.....	96
4.5A Interface.....	101
4.6A evolução do Eletrotutor.....	104

4.7O Eletrotutor IV.....	108
4.7.1A arquitetura do Eletrotutor IV.....	109
4.7.2As Estratégias de ensino e o Modelo do Aluno.....	111
4.7.3Modelo Entidade-Relacionamento (ER).....	114
5Considerações finais	119
Bibliografia.....	125

Lista de Abreviaturas

ACL	<i>Agent Communication Language</i>
AI	<i>Artificial Intelligence</i>
APPEAL	<i>A Pleasant Personal Environment for Adaptive Learning</i>
ARPA	<i>Advanced Research and Projects Agency</i>
CAI	<i>Computer Assisted Instruction</i>
CORBA	<i>Common Object Request Broker Architecture</i>
DAI	<i>Distributed Artificial Intelligence</i>
DILE	<i>Distributed Intelligent Learning Environment</i>
DLE	<i>Distributed Learning Environment</i>
DPS	<i>Distributed Problem Solving</i>
GIA	<i>Generic Instructional Architecture</i>
HTML	<i>Hypertext Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IAB	<i>The Internet Architecture Board</i>
IBASE	<i>Instituto Brasileiro de Análises Sociais e Econômicas</i>
ILE	<i>Intelligent Learning Environment</i>
ICAI	<i>Intelligent Computer Assisted Instruction</i>
ITLE	<i>Intelligent Teaching Learning Environment</i>
ITS	<i>Intelligent Tutorial System</i>
JAT	<i>Java Agent Template</i>
JDBC	<i>Java Data Base Connectivity</i>
KIF	<i>Knowledge Interchange Format</i>
KQML	<i>Knowledge Query and Manipulation Language</i>
KSE	<i>Knowledge Sharing Effort</i>
MAS	<i>Multiagent Systems</i>
MCOE	<i>Multiagent Co-Operative Environment</i>
MOO	<i>Multi User Domain Object Oriented</i>
ODBC	<i>Open Data Base Connectivity</i>
PAI	<i>Parallel Artificial Intelligence</i>
RPC	<i>Remote Procedure Call</i>
SQL	<i>Structured Query Language</i>
TCP-IP	<i>Transfer Control Protocol - Internet Protocol</i>
UFRGS	<i>Universidade Federal do Rio Grande do Sul</i>
UML	<i>Unified Modeling Language</i>
URL	<i>Uniform Resource Locators</i>
WWW	<i>World Wide Web</i>
XML	<i>Extended Markup Language</i>

Lista de Figuras

FIGURA 1 – Função de Adaptação da Metodologia	21
FIGURA 2 - Taxinomia de Agentes.....	27
FIGURA 3 - Arquitetura de Agente [TOR 95].....	29
FIGURA 4 – Comunicação e Cooperação em Sistemas Multiagentes.....	32
FIGURA 5 –Distribuição da Resolução de um Problema....	33
FIGURA 6 – Topologia de Cooperação Entre Agentes.....	35
FIGURA 7 - A Distribuição de Agentes do MCOE.....	49
FIGURA 8- Componentes do Modelo do Aluno do MCOE. 	50
FIGURA 9 - Arquitetura do Modelo Organizacional.....	69
FIGURA 10 – Fluxo de Mensagens na Fase das Lições.....	72
FIGURA 11 – Fluxo de Mensagens na Fase dos Exemplos.. 	73
FIGURA 12 – Fluxo de Mensagens na Fase dos Exercícios. 	74
FIGURA 13 – Módulos da Mensagem.....	75
FIGURA 14 – Classe KQMLMessage.....	76
FIGURA 15 – Classe ContentMessage.....	77
FIGURA 16 – Mensagem com Conteúdo Login do Usuário 	81
FIGURA 17 – Modelo Interno de Agente.....	83
FIGURA 18 - Processo de Execução do Agente Genérico....	84
FIGURA 19 – Arquitetura do Ambiente.....	87
FIGURA 20 – Arquitetura do Ambiente Eletrotutor III.....	92
FIGURA 21 – Estrutura do Agente Genérico.....	93
FIGURA 22 – Diagrama das Principais Classes.....	94
FIGURA 23 – Fluxo das Mensagens Entre os Agentes.....	100
FIGURA 24 – Tela de Abertura.....	102
FIGURA 25 - Janela de Login	102
FIGURA 26 - Interface com Conteúdo de Lições.....	103

FIGURA 27 – Interface com Conteúdo de Exemplos.....	103
FIGURA 28 – Interface com Exercício	104
FIGURA 29 – Exemplo de Aplicação de Heurísticas	106
FIGURA 30 – Tela de Abertura do Eletrotutor III.....	107
FIGURA 31 – Tela de Lição do Eletrotutor III.....	107
FIGURA 32 – Tela de Exercício do Eletrotutor III.....	108
FIGURA 33 - Arquitetura do Sistema Tutorial.....	109
FIGURA 34 – Classe Content.....	111
FIGURA 35 – Ferramenta Administrativa do JADE.....	113
FIGURA 36 - Modelo de Dados.....	114
FIGURA 37 – Processo PGP.....	122

Lista de Tabelas

TABELA 1 - Resultados das Avaliações do Eletrotutor.....	17
TABELA 2 - Taxinomia de Agentes Pedagógicos.....	41
TABELA 3 - Classificação das arquiteturas.....	41
TABELA 4 - Referências Bibliográficas sobre ILE Apontadas na TABELA 3.....	42
TABELA 5 – Comparação entre Ferramentas de Desenvolvimento.....	44
TABELA 6 – Parâmetros-Chave das Performatives.....	76
TABELA 7 - Performatives Utilizadas pelos Agentes.....	77
TABELA 8 – Conteúdo das Mensagens.....	78
TABELA 9 – Características do Eletrotutor.....	91
TABELA 10 – Casos de performance do Aluno.....	95

Resumo

Os sistemas de ensino, baseados em redes de computadores, denominados Ambientes de Ensino Distribuídos, são uma excelente alternativa de solução para o ensino a distância¹, pelo fato de manterem os recursos instrucionais centralizados, permitindo um maior controle sobre o processo, e de colocarem estes recursos simultaneamente à disposição de diversos alunos ao mesmo tempo, possibilitando oportunidades de ensino a um maior número de pessoas e diminuindo os custos de sistemas educacionais em larga escala.

Os Ambientes de Ensino Inteligentes, por outro lado, constituem uma classe de instrumentos de ensino muito mais avançados do ponto de vista pedagógico. Resultados experimentais, no entanto, levam a crer que uma solução ideal para a Educação a Distância deveria privilegiar os dois aspectos.

Os avanços mais recentes no campo dos ambientes de ensino inteligentes têm proposto o uso de arquiteturas baseadas em sociedades de agentes. Os princípios dos sistemas multiagentes têm mostrado um potencial bastante adequado ao desenvolvimento de sistemas de ensino, devido ao fato de a natureza do problema de ensino e aprendizagem ser mais facilmente resolvido de forma cooperativa. Ambientes de ensino baseados em arquiteturas multiagentes possibilitam suportar o desenvolvimento de sistemas de forma mais robusta, mais rápida e com menores custos, tornando-os mais atrativos, do ponto de vista de seu aproveitamento real, não ficando restrito a um protótipo.

O objetivo deste trabalho, portanto contempla o desenvolvimento de uma infra-estrutura de projeto, desenvolvimento e implementação de sistemas de Ambientes Inteligentes Distribuídos de Aprendizagem (*Distributed Intelligent Learning Environment* – DILE), denominado JADE – *Java Agent for Distance Education Framework* baseado em abordagem de Arquiteturas Multiagentes, voltado para a Educação a Distância.

O JADE propõe uma infra-estrutura de projeto, desenvolvimento e implementação de sistemas de Ambientes Inteligentes Distribuídos de Aprendizagem (*Distributed Intelligent Learning Environment* – DILE) baseado na abordagem de Arquiteturas Multiagentes, voltado para a Educação a Distância, para múltiplos domínios.

Palavras-Chave: Inteligência Artificial, Ambientes de Ensino Inteligentes, Informática na Educação, Educação a Distância.

¹ Apesar de, usualmente, a expressão *à distancia* empregar a crase, consagrou-se na literatura que as expressões *Ensino a Distância* e *Educação a Distância* referem-se a uma distância genérica, dispensando, assim, o seu emprego.

TITLE: “AGENTS – ORIENTED MODELING APPLIED TO INTELLIGENT LEARNING ENVIRONMENTS: JADE: JAVA AGENT FRAMEWORK FOR DISTANCE LEARNING ENVIRONMENTS”.

Abstract

Educational systems supported by computer networks - referred to as Distributed Learning Environments (DLE) - can indeed be an excellent option for overcoming distance teaching obstacles. Chiefly, because instructional resources are centralized, thus allowing for a better handle over the learning process. Secondly, these resources are concurrently available to several students at a time, subsequently opening up the doors for more learning opportunities to a greater amount of people, hence reducing the educational systems costs on a much larger scale.

On the other hand, from a pedagogical point of view, the so-called Intelligent Learning Environments (ILE) offer more advanced teaching tools. Nevertheless, experimental findings indicate that an ideal solution for Distance Education should observe both DLE and ILE methods.

The latest advances in the science of Intelligent Learning Environments (ILE) recommend the use of frameworks supported by agent associations. Advances in the multi-agent systems approach have shown a large potential for the development of educational systems, because the type of problems that emerge from teaching and learning can be solved more easily in a cooperative manner. Teaching environments managed by a multi-agent program facilitates the development of systems in a more robust and faster way, thus aptly lowering costs and making them more appealing, when you consider the real grades benefits without strictly adhering to a particular prototype.

The objective of this project therefore, is to design and build an outlined structure that will encompass an assortment of projects, developments, and implementations of Distributed Intelligent Learning Environment -(DILE) systems. This achievement is labeled as: JADE-Java Agent for Distance Education framework, which depends on a multi-agent approach conducted towards distance learning.

JADE tenders an infra-structure that is multi-faceted by projects, development, and the implementation of Distributed Intelligent Learning Environment-(DILE) systems based on a multi-agent approach directed towards multiple uses for distance learning.

Key Words: Artificial Intelligence, Intelligent Learning Environments, Computers in Education and Distance Education.

1 Introdução

A Ciência da Computação, juntamente com a Psicologia e a Educação, tem buscado aperfeiçoar ferramentas computacionais de ensino, voltadas principalmente para o ensino individualizado e, a cada dia, novas abordagens do uso da Informática na Educação têm trazido novas perspectivas para esta área.

Um importante passo foi dado com o desenvolvimento de Sistemas Tutores Inteligentes (ITS - *Intelligent Tutor System*) ou ICAI (*Intelligent Computer Assisted Instruction*), como também são conhecidos [BAR 82][VIC 89]e [CLA 90]. A proposta dos métodos de ensino por computador, até então, vinha sempre sendo voltada para o ensino programado, que propicia o aprendizado por meio de um ambiente computacional elaborado para atender a uma necessidade ou a um objetivo bem específico.

Os métodos de ensino auxiliados por computador ou CAI (*Computer Assisted Instruction*), como são conhecidos, que surgiram principalmente a partir da década de 70, nem sempre demonstram ser adequados para atender necessidades de aprendizagens mais ricas e complexas, pois limitam-se a comportar-se como meros livros eletrônicos.

Com a evolução da Informática na Educação, os ambientes computacionais de ensino passaram a ser vistos como uma excelente alternativa também para a Educação a Distância, dando a este campo da ciência um novo vigor.

A Educação a Distância tem por objetivo o desenvolvimento de ambientes e de metodologias que propiciem o aprendizado remoto, isto é, que um ou mais alunos possam vivenciar experiências de aprendizagem em local fisicamente diferente do qual o ambiente e os recursos instrucionais se encontram. A natureza das pesquisas nesta área tem caráter nitidamente multidisciplinar, unindo esforços das Ciências da Educação, da Psicologia, da Engenharia e da Ciência da Computação.

O atual estado da arte da área de Redes de Computadores, bem como a área de Multimídia, tem instrumentalizado de modo substancial o desenvolvimento de ambientes de ensino baseados em arquiteturas cliente-servidor. De modo especial a popularização da Internet, assim como o grande desenvolvimento e padronização, de fato, de protocolos e serviços, faz com que ela se torne um grande atrativo para a popularização do ensino a distância [EDW 97].

Os denominados Ambientes de Ensino Distribuídos ou *Distributed Learning Environment* DLE, são formados por uma configuração de ambiente de ensino caracterizado pela utilização de uma arquitetura distribuída ou uma arquitetura cliente-servidor. Diferentemente dos sistemas tradicionais que se caracterizam por operarem em modo *stand-alone*, isto é, autônomo e completo em si mesmo, os sistemas DLE caracterizam-se por apresentar pelo menos um componente servidor, que constitui-se no sistema propriamente dito, instalado em uma determinada plataforma e um ou mais componentes cliente, que constitui-se no programa ou programas instalados no ambiente de usuário. A comunicação entre estes componentes ocorre sobre uma arquitetura de rede que os integra por meio dos protocolos adequados.

Esta arquitetura se configura como uma excelente alternativa de solução para o ensino a distância, além de apresentar a vantagem sobre as arquiteturas tradicionais de manter os recursos de ensino de forma centralizada, permitindo um maior controle sobre

o processo, disponibilizando estes recursos simultaneamente aos diversos alunos, diminuindo os custos de implantação em larga escala.

Os ambientes distribuídos de ensino permitem colocar à disposição do aluno praticamente todos os recursos multimídia freqüentemente encontrados nos ambientes de ensino tradicionais com uma performance de qualidade e velocidade adequados, introduzindo a vantagem de o sistema poder lançar mão de recursos que não estejam fisicamente disponíveis em sua plataforma. Entretanto, o simples uso desses recursos resulta, geralmente em ambientes de ensino excessivamente estáticos e com técnicas de ensino bastante diretivas

A fim de prover aos sistemas computacionais de ensino, capacidade de adaptação ao contexto e de personalização do ambiente de acordo com as características do aluno, além de permitir um alto grau de interatividade entre o ambiente e os usuários e um controle de sessões de ensino em ambiente multi-usuários, as pesquisas apontam para o uso de recursos propiciados pela Inteligência Artificial. A introdução das técnicas de I. A. nestes ambientes tem a finalidade de propiciar mecanismos de modelagem do processo de ensino bem como do estado cognitivo do estudante [MAS 96].

Este trabalho situa-se no contexto da Informática aplicada à Educação, mais especificamente na abordagem de Inteligência Artificial, e contempla a utilização de sistemas multiagentes interconectados pela Rede Mundial de Computadores – a Internet – como instrumento de Ensino a Distância.

Para tanto, serão discutidas, a seguir, as motivações desta proposta e seus objetivos, respectivamente e, logo após, o estado da arte e os trabalhos correlatos, as premissas que nortearam este projeto e as hipóteses estabelecidas e, finalmente, os resultados obtidos e as considerações finais.

1.1 Motivação

Conforme foi discutido anteriormente, o atual estado da arte da área de Redes de Computadores, e em particular da Internet, tem contribuído muito para o desenvolvimento de ambientes de ensino baseados em arquiteturas cliente-servidor. E sua aplicação em programas de ensino a distância.

Ambientes distribuídos de ensino, baseados nesta tecnologia, permitem colocar à disposição do aluno praticamente todos os recursos multimídia freqüentemente encontrados nos ambientes de ensino tradicionais, com uma performance de qualidade e velocidade cada vez melhores, introduzindo a vantagem de o sistema poder lançar mão de recursos que não estejam fisicamente disponíveis em sua plataforma.

A utilização pura e simples de modernos recursos multimídia distribuídos na Rede, no entanto, costuma resultar, muitas vezes, na construção de ambientes de ensino estáticos e diretivos, que não condizem com as modernas teorias educacionais, pelo contrário, apresentam ao aluno, uma avalanche de material instrucional, deixando totalmente sob a sua responsabilidade a ação a ser tomada no processo de aprendizagem.

De outro lado, o campo da Inteligência Artificial em Educação, por sua vez, tem experimentado um avanço dos sistemas tutoriais inteligentes na direção de vários tipos de sistemas denominados Ambientes Inteligentes de Aprendizagem (ILE- *Intelligent Learning Environment*). Esta tendência tem tido razões tanto computacionais quanto educacionais, isto é, pode ser explicada em termos de uma nova compreensão da

Psicologia da Aprendizagem, da reconhecida dificuldade da construção de sistemas tutoriais e do apelo da abordagem de Ensino Centrado no Aluno [SEL 92].

Segundo [ROE 94], cabe a um ILE, as tarefas de capturar uma amostra suficientemente larga das características cognitivas do aluno, analisar descrever e indexar a maneira mais apropriada para converter esta amostra em um conjunto de experiências de aprendizagem coerentes e cumulativas dirigindo uma participação pró-ativa ou reativa, gerar experiências de aprendizagem, perceber estas experiências em termos de parâmetros de uma interface apropriada ao aluno; e identificar e criar sistemas de resgate das experiências de aprendizagem.

Esta abordagem considera que existem diferentes classes de alunos: os alunos denominados *Não-cooperativos*, que atuam de modo passivo ou, até mesmo tentam frustrar os objetivos do professor, os denominados *Cooperativos*, que seguem as orientações de seus professores mas não necessariamente sabem aonde ir; e os alunos *Pró-ativos*, que sabem muito bem seus objetivos, e buscam assistência para tornar o peso da tarefa o mais leve possível. Obviamente a metodologia empregada em cada um destes três casos é diferente e deve haver uma preocupação clara de parte do ILE a respeito do tipo de aluno para o qual ele foi projetado para assistir.

Para isto, torna-se necessária uma modelagem de usuário, que deve partir para uma clara especificação do mesmo. O ambiente de ensino inteligente deve construir e atualizar o modelo de aluno em função do que o aluno já sabe, o que pode variar significativamente de um aluno para outro. Esta diferença deve ser devidamente considerada na busca da eficiência, quando se desenvolve ambientes de ensino inteligentes, e considerar variáveis como a experiência prévia, a perícia do aluno no domínio considerado, a transparência da terminologia técnica, os propósitos do aluno, sua experiência prévia e suas expectativas.

Assume-se, portanto, a hipótese de que, se, por um lado, os denominados Ambientes de Ensino Distribuídos ou *Distributed Learning Environment* DLE, que privilegiam a utilização de uma arquitetura distribuída em rede, configuram-se como uma excelente alternativa de solução para o ensino a distância, por manter os recursos de aprendizagem de forma centralizada e permitir um maior controle sobre o processo, disponibilizando estes recursos simultaneamente a diversos alunos, diminuindo os custos de implantação em larga escala, os ILE constituem uma classe de instrumentos de ensino muito mais avançados do ponto de vista pedagógico, e que uma solução ideal para a Educação a Distância deveria privilegiar os dois aspectos.

Esta hipótese é reforçada pelo resultado de trabalhos anteriores, nos quais, com a finalidade de verificar a eficácia de cada uma destas duas abordagens de sistema tutorial foram elaboradas, duas avaliações de diferentes ambientes de ensino, através de pesquisa experimental. [SIL 97a].

Um ambiente de ensino denominado Eletrotutor foi implementado no contexto de uma dissertação de mestrado [SIL 92] e [SIL94], com o objetivo de servir de instrumento de pesquisa cujo propósito foi verificar a eficácia do uso de ambientes tutoriais inteligentes na escola. Este sistema aborda uma unidade de estudo no campo do ensino da Física, mais precisamente da Eletricidade, no estudo da Lei de Ohm e suas aplicações.

Posteriormente, uma segunda versão foi implementada [SIL 96b], também com objetivos de avaliação de efetividade como ferramenta educacional, porém sob outro

paradigma. O conteúdo do Eletrotutor é constituído por oito unidades, formadas por lições, exemplos e exercícios.

A primeira versão do Eletrotutor foi implementada em linguagem ARITY-PROLOG. A sua arquitetura seguiu, com algumas adaptações, o modelo de sistemas tutores inteligentes proposto por [VIC 89].

O Eletrotutor II, por sua vez, foi desenvolvido em linguagem HTML e *JavaScript*, com uma interface mais agradável e propiciando ao aluno o controle do processo através de um menu. Para cada uma das unidades foi desenvolvido um conjunto de páginas HTML contendo os textos e as figuras que constituem o conteúdo apresentado pelo Eletrotutor. A implementação do Eletrotutor II [SIL 97] procurou preservar ao máximo as características funcionais da versão anterior. Foi mantida a estrutura de pré-requisitos e de conteúdo instrucional, assim como as principais estratégias de ensino utilizadas. Foi empregado o mesmo sistema de apresentação de conteúdos por meio de *frames* com textos e figuras, seguido da apresentação de exemplos gerados pelo sistema e da proposição de exercícios. Sem, no entanto, exercer nenhum controle sobre o processo de ensino.

Os resultados são mostrados, de forma sucinta, no quadro da TABELA 1, a seguir, com um resumo dos dados.

Os alunos que tiveram aula no Eletrotutor tiveram média 7,80 e os alunos que assistiram aula expositiva tiveram média 8,38, sendo a diferença de 0,58.

Na segunda pesquisa, os alunos que assistiram aula no Tutor II, tiveram média 7,90 e os alunos que assistiram aula expositiva tiveram média de 8,51, sendo a diferença de 0,61 a favor da aula expositiva.

Verifica-se, assim, que nos dois casos a média obtida pelos alunos da aula expositiva é um pouco superior à dos alunos que utilizaram os ambientes e que estas diferenças se mantiveram relativamente constantes nas duas pesquisas. Para obter maiores detalhes deste experimento ver [SIL 97].

TABELA 1 - Resultados das Avaliações do Eletrotutor.

		N° de alunos	Médias	Difer. das médias	Desvio-Padrão	Valor das Estatísticas	P	Significância
Eletrotutor I (1992)	Tut I	15	7,80	0,58	1,32	Teste T de Student calculado T=2,29	0,26	Não-Significativo
	Expo I	13	8,38		1,39	Z=1,33 U de Mann-Whitney calculado	0,18	
Eletrotutor II (1997)	Tu II	73	7,90	0,61	1,51	Teste T de Student calculado T= 2,58	0,011	Significativo
	Expio II	76	8,51		1,36	Z= 2,54 U de Mann-Whitney calculado	0,011	

Nas duas avaliações experimentais (1992 e 1997) os achados demonstraram diferenças a favor da aula expositiva. Porém com a primeira versão do Eletrotutor essa diferença, de acordo com o tratamento estatístico, não chega a ser significativa, o que não ocorre com a segunda versão, onde essa diferença mostrou-se muito significativa.

A primeira versão do Eletrotutor (1992), com a utilização de técnicas de Inteligência Artificial, exerce um controle sobre a quantidade mínima de exemplos e exercícios que o aluno fazia monitorando o seu desempenho, tornando mais eficaz o ensino, como podemos ver através da análise dos resultados onde a aula tutorial proporcionava notas tão boas quanto às das aulas expositivas, não havendo diferença significativa entre as notas obtidas pelos alunos dos dois grupos. Na segunda versão do Eletrotutor, este controle não ocorre, pois não foi implementada nenhuma forma de modelagem de aluno ou de adaptação da estratégia de ensino.

Os resultados levam a crer que a simples mudança de paradigma de um tutor inteligente para uma arquitetura cliente-servidor, e o emprego de uma interface gráfica, não são suficientes para obter resultados, pelo menos equivalentes, ao de uma aula expositiva com a presença do professor, pois na versão anterior, que exerce um certo grau de controle tutorial, com uma ação pedagógica mais diretiva, mesmo considerada não tão atrativa do ponto de vista da interface, sem nenhum outro recurso da multimídia, foi, em 1992, mais bem sucedido na eficácia do ensino, pois as notas igualaram-se estatisticamente, as do Eletrotutor I com as da aula expositiva.

Embora deva se registrar que estes trabalhos não tenham sido feitos com o rigor científico suficiente, as conclusões obtidas nestes experimentos juntamente com alguns fatores, descritos a seguir, levantados a partir da análise da literatura, incentivaram o desenvolvimento deste trabalho.

Boa parte dos trabalhos de desenvolvimento de ambientes de ensino permaneceram restritos aos ambientes de laboratório, devido principalmente às seguintes razões:

- Várias tecnologias têm sido desenvolvidas no que se refere ao desenvolvimento de ambientes de ensino mas dificilmente elas são agregadas em um projeto único que aproveite o melhor do estado da arte de cada uma delas e que possa evoluir à medida que cada uma delas evolui;
- Os ambientes de ensino são tipicamente desenvolvidos como aplicações customizadas, ou seja, projetos *ad-hoc*, feitos sob medida para os fins a que se propõem;
- Os desenvolvedores de ambientes de ensino raramente utilizam ferramentas de desenvolvimento, o que permitiria uma melhor documentação das etapas de desenvolvimento deles e faria com que os avanços obtidos pudessem ser reaproveitados mais facilmente em projetos similares;
- Os ambientes de ensino raramente são desenhados para evoluir, e o desenvolvimento de ambientes de ensino não costumam ser compreendidos como um processo de refinamento iterativo;
- Em geral, o projeto de ambientes de ensino não incorpora produtos comerciais de software;

Os avanços mais recentes no campo dos ambientes de ensino inteligentes têm, conforme foi visto, proposto o uso de arquiteturas baseadas em sociedades de agentes. Os princípios dos sistemas multiagentes têm mostrado um potencial bastante adequado ao desenvolvimento de sistemas de ensino, devido ao fato de a natureza do problema de ensino e aprendizagem ser mais facilmente resolvido de forma cooperativa.

Ambientes de ensino baseados em arquiteturas multiagentes possibilitam suportar o desenvolvimento de sistemas de forma mais robusta, mais rápida e com menores custos, tornando-os mais atrativos, do ponto de vista de seu aproveitamento real, não ficando restrito a um protótipo.

1.2 Objetivos, premissas e hipóteses

A partir da motivação anteriormente descrita, e do levantamento teórico realizado, foram determinados os objetivos e os preceitos que serão discutidos a seguir, para o desenvolvimento deste projeto:

1.2.1 Objetivo geral

O objetivo deste trabalho é propor e discutir uma infra-estrutura de projeto, desenvolvimento e implementação de sistemas de Ambientes Inteligentes Distribuídos de Aprendizagem (*Distributed Intelligent Learning Environment – DILE*) baseado na abordagem de Arquiteturas Multiagentes, voltado para a Educação a Distância, para múltiplos domínios.

Para tanto foram estabelecidos os seguintes objetivos específicos:

1.2.2 Objetivos específicos

- propor uma arquitetura de uma sociedade de agentes, a partir de uma metodologia de modelagem de sistemas orientada a agentes, que possa ser aplicada para o projeto de ambientes de ensino, em diversos domínios, de modo coerente com os demais objetivos estabelecidos;
- propor uma arquitetura interna de agente compatível com a metodologia de ensino, com o modelo da sociedade de agentes estabelecida e com a metodologia de modelagem adotada;
- propor a utilização de mecanismos de comunicação e de cooperação entre agentes adequado a uma sociedade de agentes pedagógicos;
- propor avanços na modelagem de estados mentais internos dos agentes que vem sendo adotada pelo grupo de pesquisas e no estado da arte;
- avaliar o atendimento dos objetivos implementando um DILE para avaliar a arquitetura proposta;
- contribuir para as pesquisas que visam o estabelecimento de uma metodologia e de técnicas de ensino apropriadas e possíveis de serem implementadas, dentro das premissas estabelecidas.

Para alcançar esses objetivos, algumas hipóteses e alguns pressupostos foram considerados, baseados em trabalhos anteriores e no estado da arte. Estas premissas serão discutidas a seguir.

1.2.3 Premissas e hipóteses sob o ponto de vista computacional

Do ponto de vista computacional, acredita-se que os processos de ensino e de aprendizagem devem ser modelados e tratados como um sistema complexo envolvendo basicamente a interação do indivíduo com o seu meio ambiente, principalmente com outros indivíduos, exercendo diferentes papéis. Nem os indivíduos nem o meio ambiente podem ser encarados de forma passiva no processo e as interações que ocorrem são de natureza dinâmicas e complexas.

Portanto, nos processos de ensino e de aprendizagem, considera-se impossível estabelecer todas as variantes ou prever todas as situações envolvidas. Por isso, um modelo computacional adequado não consegue modelar, previamente, todas as situações. Principalmente quando se trata de um ambiente no qual muitos usuários podem estar atuando simultaneamente.

Neste trabalho, assume-se que o modelo computacional de um ambiente de ensino deve ser elaborado como um problema a ser solucionado de forma cooperativa e sinérgica, ou seja, os objetivos educacionais devem ser encarados como objetivos ou metas a serem atingidas por uma sociedade de agentes inteligentes composta por indivíduos e programas.

Este projeto baseia-se, assim, na hipótese de que os ambientes de ensino podem efetivamente ser decompostos em coleções de agentes independentes, ou de diferentes papéis que um agente pode desempenhar, e que trocam informações e cooperam mutuamente, usando uma linguagem formal expressiva e, além disso, tal decomposição promove modularidade, robustez e interoperabilidade. O processo de aprendizagem

resulta, assim, da sinergia entre agentes pedagógicos, especializados em diferentes funções, e sua interação com os alunos.

Mais especificamente, deve possuir uma arquitetura distribuída em rede, através de um sistema composto de uma sociedade de agentes inteligentes com capacidade de comunicação e cooperação. O atual estado da arte dos serviços e protocolos de rede disponíveis possui uma ampla variedade de recursos que podem ser utilizados com propósitos didáticos em ambientes inteligentes de ensino. Estes recursos devem, no entanto ser empregados de modo integrado e adequado à metodologia de ensino adotada.

1.2.4 Premissas e hipóteses sob o ponto de vista educacional

A atividade educacional proposta por um ambiente computacional que se propõe a ser uma alternativa para um problema tão complexo como a Educação a Distância não deve incorrer nos mesmos equívocos tão comuns no ensino tradicional, assim como deve levar em conta suas peculiaridades, especialmente, no que se refere à natureza dos seus objetivos e às características da sua clientela, predominantemente constituída por jovens ou adultos com formação diversificada. Deve-se, portanto, ter em mente alguns pressupostos que o diferenciam do ensino tradicional.

À parte de discussões paradigmáticas de qual o modelo de ensino a ser empregado em um ambiente computacional, que será abordado mais adiante, este projeto baseia-se na hipótese de que um diagnóstico cognitivo acurado do aluno usuário de um ambiente de ensino, denominado, aqui, Modelo do Aluno, deve levar em conta os seguintes indicadores:

- **Conhecimentos:** diz respeito àquilo que o aluno já conhece sobre o conteúdo a ser ensinado e o que ele vai passando a conhecer na medida em que ele utiliza o ambiente;
- **Habilidades:** diz respeito às aptidões relacionadas ao conteúdo propriamente dito, que são os pré-requisitos que o aluno deve possuir para estar em condições de executar as atividades propostas, que habilidades de conteúdo necessárias ao acompanhamento do curso o aluno deve possuir e quais as que ele possui num determinado momento. Além disso, deve ser levado em conta as habilidades relacionadas ao uso das ferramentas disponíveis no ambiente de ensino: que habilidades o aluno necessita para utilizar as ferramentas e os recursos que o sistema disponibiliza a ele e quais as que ele realmente possui. Nada adianta, por exemplo, colocar ferramentas de *e-mail*, *chat*, *www*, e outros, se o aluno nunca as utilizou e o objetivo dele passa a ser aprender a utilizá-las em vez do objetivo principal do curso;
- **Atitudes:** conforme foi discutido anteriormente, pode-se, de um modo geral, dividir os alunos, no que diz respeito a suas atitudes diante do uso do ambiente de ensino, em três grupos: *não-cooperativos*, *cooperativos* e *pró-ativos*. Um aluno não-cooperativo, por exemplo, poderia utilizar uma ferramenta de *chat* ou o *browser* para brincar ou navegar pela Internet. Um aluno cooperativo utilizaria estas ferramentas de forma correta e participaria de discussões. Um aluno pró-ativo não só as utilizaria como também tomaria atitudes de liderança no grupo propondo, por exemplo, discussões, indicando material complementar. Um diagnóstico de modelagem de aluno deve buscar

formas de identificar e tentar quantificar estes indicadores e utiliza-los adequadamente na metodologia de ensino empregada.

A figura 1, abaixo, mostra esta idéia de uma forma gráfica:

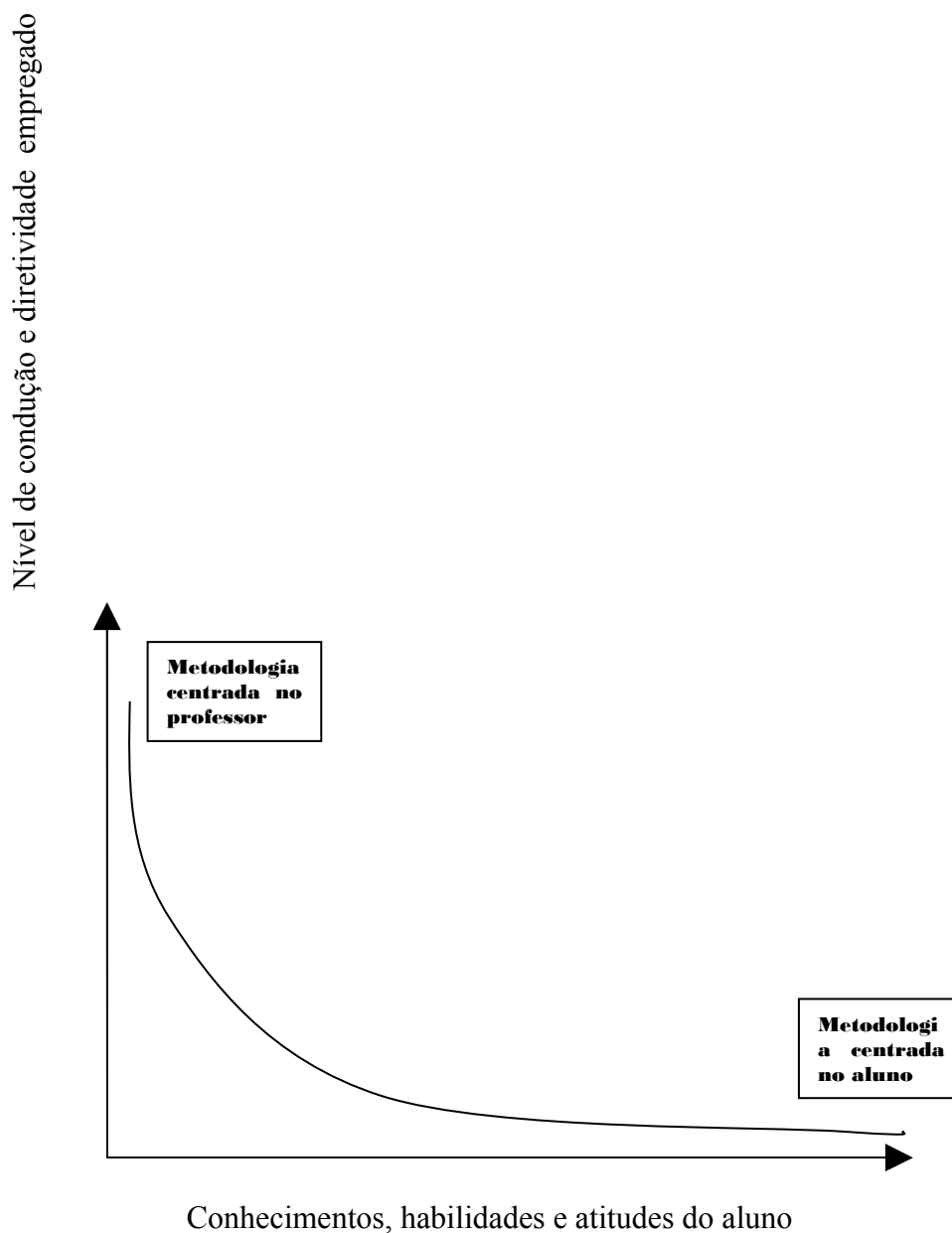


FIGURA 1 – Função de Adaptação da Metodologia

Na medida em que o ambiente percebe que o aluno tem uma característica que se aproxima mais do extremo esquerdo do gráfico, a metodologia de ensino empregada deve ser mais diretiva e mais condutista, onde o professor, representado pelo ambiente de ensino, “dá as cartas” dirigindo o processo de uma forma mais determinada. Por outro lado, conforme o sistema percebe que o aluno se aproxima do outro extremo, adotando uma atitude pró-ativa e possuindo as habilidades e os conhecimentos

necessários, o sistema deve se adequar, passando a desempenhar o papel de facilitador, em uma metodologia de ensino totalmente centrada no aluno.

A função do Modelo do Aluno é procurar manter um diagnóstico *on the fly*, ou seja, enquanto a sessão de ensino transcorre da forma mais acurada possível, através de uma quantidade adequada de informações, levantadas a partir da interação do aluno com o ambiente. O atual estado da arte mostra que os ambientes de ensino ainda não provêm este recurso de forma satisfatória.

No item, a seguir, serão discutidos o estado da arte e alguns trabalhos correlacionados, referentes aos ambientes de ensino e às ferramentas de desenvolvimento de sistemas multiagentes.

2 Estado da arte e trabalhos correlatos

Neste capítulo serão discutidos basicamente os conceitos fundamentais e os avanços nos campos da Educação a Distância e em Inteligência Artificial, que nortearam este projeto.

2.1 Estado da arte em Educação a Distância

Há uma nova visão de Educação, desenvolvida nos últimos 15 a 20 anos, influenciada fortemente pelo avanço da Ciência Cognitiva. O sistema educacional vem sendo cada vez mais focado sobre a aprendizagem em vez do ensino. O desenvolvimento de teorias da aprendizagem tem mudado a natureza do aprendizado e da percepção do aluno. O conhecimento é hoje considerado como algo socialmente construído através da ação, da comunicação e da reflexão por parte dos alunos.

A visão clássica de ensino através da transmissão de conhecimento vem evoluindo para um modelo de experimentação prática e de interação que promove mudanças nos conceitos e estratégias dos alunos, até a atingir a proficiência. Neste contexto, os professores passam a desempenhar um papel de facilitadores em vez de provedores de informação. A Educação a Distância, no seu atual estado da arte, quando levada de forma correta, aplica-se muito bem a esta concepção de Educação.

Conforme foi abordado em trabalho anterior [SIL 98], a idéia de Educação a Distância, embora seja antiga, tem mostrado grande capacidade para integrar novas tecnologias, com sucesso. Nos anos mais recentes, de um modo especial, tem havido uma grande explosão dos mecanismos e ferramentas disponíveis para a sua implementação e suporte.

As definições clássicas de ensino à distância implicam, em geral, que a situação ideal de aprendizagem é a tradicional, com o professor e o aluno estando face-a-face. A Educação a Distância, neste ponto de vista, seria uma forma “inferior” de educação, tentando sempre preencher as deficiências deste modelo. Esta concepção pode ser verdadeira em muitos casos, mas está havendo um crescente corpo de pesquisas que exploram outras opções, especialmente à luz do desenvolvimento de novos paradigmas educacionais, das mudanças na dinâmica social, e no avanço tecnológico dos meios de comunicação e computacionais.

Para [NUN 97], é importante observar que a Educação a Distância não pode ser vista como substitutiva da educação convencional, presencial. São duas modalidades do mesmo processo. A Educação a Distância não concorre com os meios convencionais, tendo em vista que não é esse o seu objetivo. Se a Educação a Distância apresenta como característica básica a separação física e temporal entre os processos de ensino e aprendizagem, isto significa não somente uma qualidade específica dessa modalidade, mas essencialmente, um desafio a ser vencido, promovendo-se de forma combinada, o avanço na utilização de processos cooperativos de ensino.

Keegan [KEE 86] resume os elementos centrais, considerados neste projeto, que caracterizam os conceitos de Educação a Distância:

- Separação física entre professor e aluno, que a distingue do Ensino Presencial;

- Influência da organização educacional (planejamento, sistematização, plano, projeto e organização dirigida), que a diferencia da Educação Individual;
- Utilização de meios técnicos de comunicação para unir o professor ao aluno e transmitir os conteúdos educativos;
- Disponibilidade de uma comunicação de mão dupla, onde o estudante se beneficia de um diálogo e da possibilidade de iniciativas de dupla via;
- Possibilidade de encontros ocasionais com propósitos didáticos e de socialização;
- Participação de uma forma industrializada de educação, a qual, se aceita, contém o germe de uma radical distinção dos outros modos de desenvolvimento da função educacional.

[SPO 95]aponta, ainda, uma definição mais simples e abrangente, que explora novas possibilidades:

"Educação a Distância é um sistema que deve prover toda e qualquer oportunidade educacional que seja necessária para qualquer um, em qualquer lugar, e a qualquer tempo. "

Existem denominações que, por vezes, se confundem e, por outras, são empregadas com significados diversos: Teleducação, Ensino a Distância, Aprendizagem a Distância, Educação Aberta são diferentes enfoques. De um modo geral, todas as denominações implicam em oferecer oportunidades de ensino, de forma aberta e democrática, a diferentes segmentos da população, acomodando diferentes situações e necessidades.

O fato é que, mais importante do que a denominação, a base teórica dos modelos instrucionais afeta não só a forma como a informação é comunicada ao aluno, mas também a forma como o aluno entende e constrói um novo conhecimento a partir das informações apresentadas. [SHE 96]destaca duas diferentes abordagens que influenciam os diversos projetos instrucionais de Educação a Distância:

A primeira delas é conhecida como abordagem de Processamento de Símbolos (*symbol-processing*) [EYS 94][SIM81]. Esta abordagem é baseada no conceito de que o conhecimento é constituído por operações formais com símbolos. O conceito-chave é que o professor pode transmitir um corpo fixo de informações aos alunos por meio de uma representação externa. Representa uma idéia abstrata através de uma representação concreta e então a apresenta ao aluno através de algum meio. O aluno, por sua vez, compreende, decodifica e armazena a representação.

Uma idéia mais avançada desta teoria adiciona dois novos fatores: o contexto do aluno (ambiente, situação corrente e outras entradas sensoriais) e o intelecto (memórias, associações, emoções, interferências e raciocínios, curiosidades e interesse). O aluno então desenvolve sua própria representação e a usa para construir o novo conhecimento, em um contexto, baseado em seu conhecimento anterior e nas suas habilidades.

A segunda abordagem, denominada Cognição Situada (*situated cognition*) ou como aprendizagem baseada na solução de problemas (*problem-based learning*), adota o princípio do construtivismo [MIZ 86][PIA 77], no qual o aluno ativamente constrói uma representação interna do conhecimento através de interação com o material a ser aprendido. De acordo com este ponto de vista, a interação social e física entram na definição de um problema e na construção da sua solução. Nem a informação a ser aprendida nem a sua descrição simbólica é especificamente colocada fora do processo de questionamentos e conclusões que emergem do processo. O foco do processo é

deslocado da forma tradicional de transmissão de conhecimentos para outra mais interativa, complexa e envolvente.

Já [PET 98]aponta três principais formas de aprendizagem identificadas no ensino a distância: Aprendizagem Dialogal, Aprendizagem Estruturada e Aprendizagem Autônoma.

De acordo com [SPO 95], cinco pontos fundamentais são apontados como essenciais para que um programa de Educação a Distância atinja satisfatoriamente seus objetivos, aos quais procurou-se atender neste projeto:

- contato entre o professor e o aluno,
- aprendizagem ativa por meio de respostas do aluno,
- realimentação rápida do grau de compreensão do aluno, ao professor,
- realimentação rápida ao aluno, de seu próprio desempenho;
- oportunidade de o aluno fazer revisões e aprender com seus próprios erros.

No nível micro, o maior problema com a implementação de sistemas tradicionais de ensino auxiliados por computador é a sua falta de habilidade em prover um ensino individualizado, adaptado ao aluno. Diferentes abordagens na Informática em Educação têm sido apresentadas na literatura (ver [SIL 96a], [SIL97]e [SIL97a]).

Na abordagem de ITS [VIC 89], a performance do aluno é interpretada em termos do modelo de representação do sistema especialista e, por isso, o grau de adaptabilidade dos ITS é limitada, e os modelos de ITS clássicos não são robustos o suficiente para prover os requerimentos mínimos necessários para um ambiente interativo de ensino, que segundo [MAS 96]devem ser: *interatividade, adaptabilidade, robustez, monitoramento da aprendizagem, avaliação empírica e parcimônia.*

Um *Intelligent Learning Environment* – ILE [SEL 92], por outro lado, proporciona, ao estudante, um conjunto de experiências de aprendizagem coerentes e cumulativas dirigindo uma participação pró-ativa ou reativa. O objetivo da experiência de aprendizagem proposta é habilitar o aluno a adquirir um nível de confiança onde a situação de vivência real torne-se para o aluno um ambiente de aprendizagem. Para que isto ocorra, alguma forma de realidade virtual deve ser provida ao aluno, com os devidos limites exigidos por medida de saúde, uma vez que essas tecnologias devem ser utilizadas com parcimônia.

2.2 Estado da arte em IA.

O campo da Inteligência Artificial, (A. I.) vem se desenvolvendo no sentido da Inteligência Artificial Distribuída (*Distributed Artificial Intelligence* - DAI), como uma sub-área de A. I. que estuda a solução de problemas cooperativos através de um grupo descentralizado de processos ou agentes [DEM 93]. Estes agentes costumam ser acoplados entre si, mas mutuamente independentes, com capacidade para raciocínio, planejamento e comunicação.

As pesquisas na área de agentes, entretanto, são muito vastas e abrangentes, podendo ser divididas, segundo [BRA 97], em duas linhas principais, a primeira delas começou por volta de 1977, e a Segunda, em 1990.

A primeira linha, cujas raízes são essencialmente no campo da Inteligência Artificial Distribuída, concentra-se principalmente em desenvolver agentes deliberativos, construídos com modelos simbólicos internos. Esta abordagem tem contribuído na compreensão dos problemas “macro”, tais como a interação e a

comunicação entre agentes, a decomposição e a distribuição de tarefas, coordenação, cooperação, resolução de conflitos através de negociação, entre outros.

A segunda corrente de pesquisas, por outro lado, é um movimento mais recente e que vem crescendo rapidamente, propõe-se a estudar um conjunto de tipos de agentes muito mais abrangentes, a que [BRA 97] refere-se como desde os “débeis mentais” aos “moderadamente espertos”. O enfoque desloca-se da deliberação para o fazer, e do raciocínio para a ação remota. A grande diversidade de aplicações e abordagem tem feito das pesquisas em sistemas multiagentes uma das principais correntes em Inteligência Artificial.

Em trabalho anterior, [SIL 98] foram aprofundados alguns conceitos sobre a abordagem de agentes de software e de arquiteturas multiagentes, essencialmente no contexto da Inteligência Artificial Distribuída, e sua aplicação como solução no desenvolvimento de ambientes interativos inteligentes de ensino auxiliados por computador. Alguns pontos importantes serão resumidos, aqui, a fim de transmitir uma idéia do contexto do projeto.

Segundo [TOR 95], a Inteligência Artificial Distribuída - DAI, divide-se em três áreas: *Distributed Problem Solving (DPS)*, *Multi Agent Systems (MAS)* e *Parallel Artificial Intelligence (PAI)*. **DPS** interessa-se pela decomposição de problemas entre um número de módulos que cooperam entre si e dividem conhecimento, desenhados especificamente para um problema em particular. **MAS**, por sua vez, caracteriza-se pela existência de um certo número de agentes autônomos, heterogêneos e potencialmente independentes, trabalhando juntos para resolver um problema. Esses agentes são aptos a se adaptarem ao meio ambiente, reagir a ele e provocar mudanças neste meio. E, por último, **PAI** interessa-se mais por problemas de performance do que por avanços conceituais, preocupando-se principalmente em desenvolver linguagens e algoritmos de computação paralela.

[BRA 97] descreve um agente como uma entidade de software que funciona de forma contínua e autônoma em um ambiente em particular, geralmente habitado por outros agentes, e que seja capaz de intervir no seu ambiente, de forma flexível e inteligente, sem requerer intervenção ou orientação humana constantes. De um modo ideal, um agente que funcione, continuamente, por longos períodos de tempo, deve ser capaz de aprender com a experiência e, se ele habita um ambiente com outros agentes, seja capaz de comunicar-se e cooperar com eles, e ainda mover-se de um local para outro.

Este conceito, adotado neste trabalho, caracteriza um modelo ideal de agente que deve ser visto como um conceito que abarca uma família de outros tipos de agentes mais específicos e limitados. Um agente pode, portanto, possuir, em maior ou menor grau, os seguintes atributos:

- *Reatividade*: A habilidade de perceber o ambiente de modo seletivo e manifestar um comportamento como resposta a um estímulo externo;
- *Autonomia*: Comportamento dirigido a objetivos, pró-ativo e auto-iniciado;
- *Comportamento cooperativo*: Trabalhar com outros agentes para atingir um objetivo comum;
- *Habilidade de comunicação ao nível de conhecimento*: Capacidade de comunicar-se com pessoas ou outros agentes em uma linguagem de mais alto nível que um simples protocolo de comunicação programa a programa;
- *Capacidade de inferência*: Capacidade de agir a partir de especificações abstratas de tarefas, usando conhecimentos prévios;

- *Continuidade temporal*: Persistência de identidade por longos períodos de tempo;
- *Personalidade*: Capacidade de demonstrar atributos de um personagem;
- *Adaptabilidade*: Habilidade de aprender com a experiência;
- *Mobilidade*: Habilidade de migrar de uma plataforma para outra.

A figura 2, a seguir apresenta uma taxinomia de agentes definida em termos de um espaço em três dimensões: Grau de Mobilidade, Grau de Inteligência e Grau de Agenciamento, adaptada de [BRA 97]. Grau de agenciamento é a medida de autonomia e autoridade que o agente possui; grau de inteligência é a medida de raciocínio e capacidade de aprendizagem; e grau de mobilidade é a medida da sua capacidade de se deslocar através da rede.

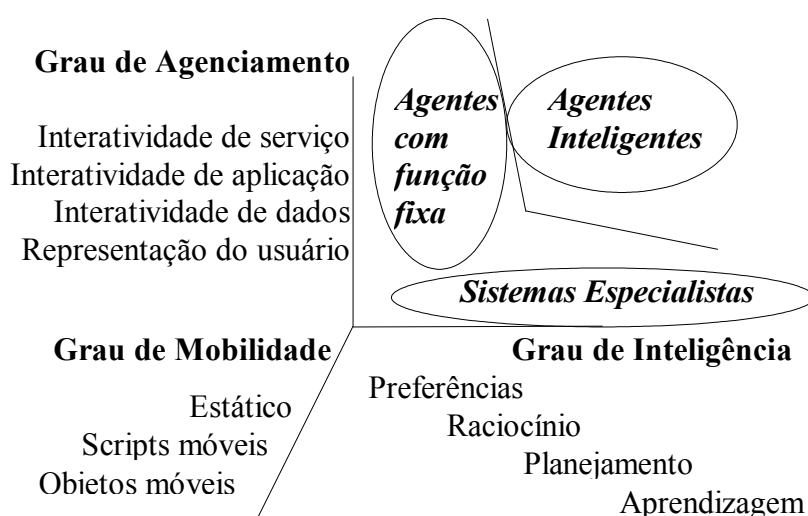


FIGURA 2 - Taxinomia de Agentes.

A forma como foram concebidos os agentes neste projeto, contempla uma categoria de agentes que poderia ser enquadrado com um elevado grau de agenciamento, pelo fato de possuírem um sistema interativo bastante complexo e um razoável grau de inteligência, com um sistema de raciocínio sem, no entanto, contemplar aspectos de aprendizagem por parte dos agentes. Os agentes do JADE não possuem mobilidade, pois não foi detectado até o presente momento tal necessidade.

Sob o ponto de vista da constituição de uma Sociedade de Agentes, a fundamentação dos sistemas multiagentes é baseada na interação social de indivíduos que convivem entre si e interagem mutuamente para alcançar objetivos comuns e individuais. Para tal, um agente é concebido como um indivíduo autônomo, com capacidades que lhes são inerentes para o desempenho de suas funções e o alcance dos seus objetivos.

Uma sociedade de agentes divide um mundo comum, e cada um dos membros desta sociedade possui diferentes objetivos e pontos de vista, gerando muitas vezes alguns conflitos. Esses conflitos devem ser negociados e resolvidos entre os agentes, e devem estar comprometidos com um plano conjunto. Este plano é um conjunto de compromissos de ações e de crenças em diferentes níveis de abstrações.

Uma sociedade de agentes, segundo [TOR 95], para atingir objetivos comuns, deve ser constituída por elementos capazes de desempenhar as seguintes funções:

- Cooperação.
- Resolução de Conflitos.
- Negociação.
- Comprometimentos.
- Interação.
- Comunicação.

A sociedade de agentes que constitui o JADE contempla, principalmente, aspectos de cooperação, comprometimentos, interação e comunicação.

Os Sistemas Multiagentes podem ser subdivididos em duas abordagens principais: os denominados *Sistemas Multiagentes Cognitivos*, cuja característica principal é a existência de uma forma explícita de representação de conhecimento, e os *Sistemas Multiagentes Reativos*, cuja ênfase principal é no comportamento, sem uma preocupação maior com a representação do conhecimento.

Enquanto que os pesquisadores da área dedicada aos estudos de multiagentes cognitivos trabalham baseados na explícita representação do conhecimento, a abordagem de sistemas multiagentes reativos é baseada na representação do comportamento. Esta abordagem [WAV 92] defende que a atividade de um agente é produzida pela interação entre o agente e seu meio ambiente, e não pelo processo de raciocínio que ocorre internamente. Segundo esta abordagem, é possível, através da representação do seu comportamento, modelar o agente, seu ambiente e suas interações causais.

Os agentes cognitivos, categoria na qual se enquadram os agentes do JADE, por outro lado, possuem uma representação formal explícita do ambiente e dos outros agentes. Podem, portanto, comunicar-se entre si e planejar sua ação futura, com uma capacidade de raciocínio sobre uma representação formal, interna, do conhecimento de que dispõem.

O modelo de sociedade de agentes utilizada pelo JADE, bem como a arquitetura interna dos agentes serão apresentados adiante, no capítulo 3. Estes modelos utilizam conceitos obtidos a partir da definição de uma arquitetura genérica de agente apresentada por [TOR 95]. Esta arquitetura compreende diferentes níveis, conforme mostrado na figura 3, e é definida como um número de componentes que representam o estado mental de um agente e a interação entre eles.

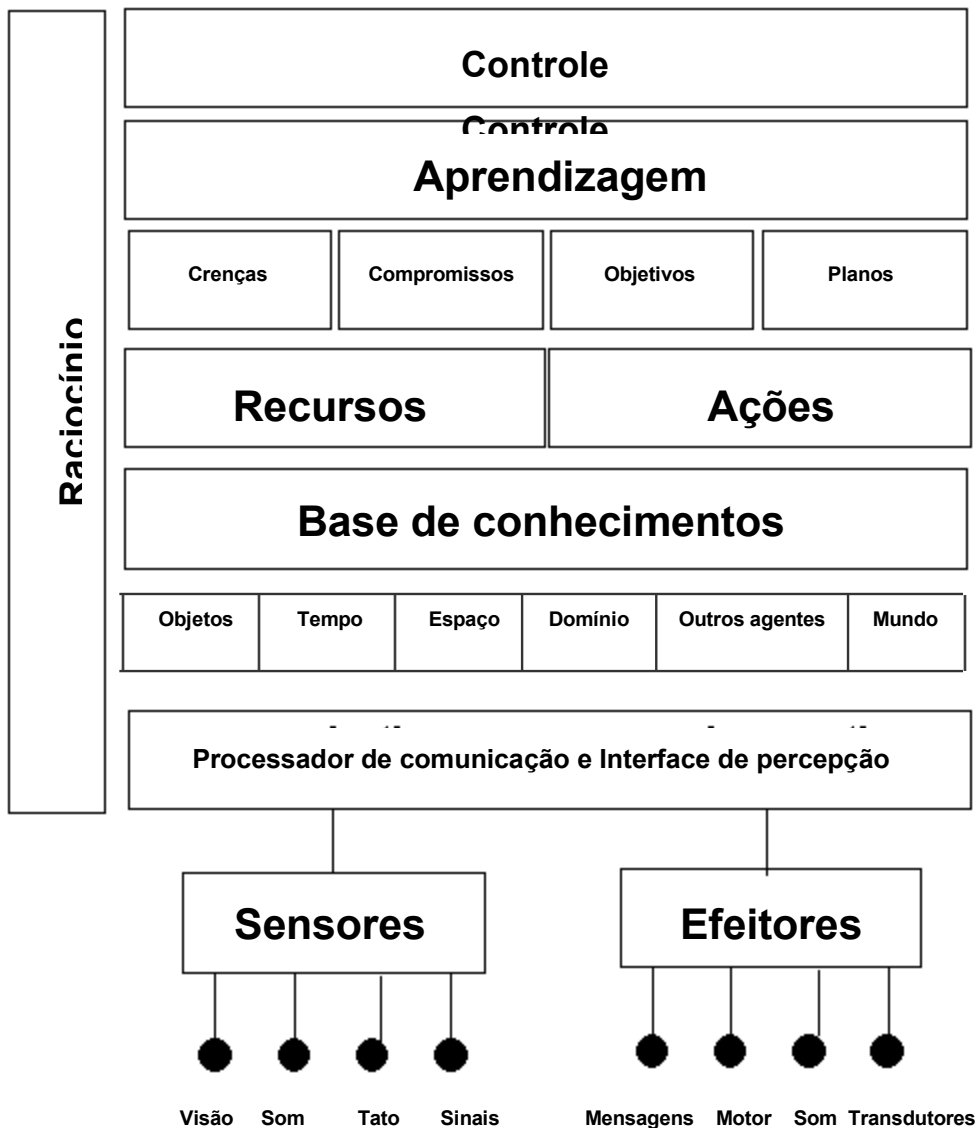


FIGURA 3 - Arquitetura de Agente [TOR 95].

Os componentes apresentados na arquitetura de [TOR 95] têm as seguintes funções:

- **Sensores:** são o meio através do qual, um agente percebe seu mundo; inclui visão, som e sinais, na forma de informações de entrada para o agente.
- **Efeitores:** são o meio pelo qual o agente afeta seu mundo, seja movendo objetos de um lugar para outro ou enviando mensagem a outro agente. Formam os sinais de saída do agente.
- **Processador de comunicação e interface de percepção:** é o componente através do qual as informações de entrada e saída são analisadas, classificadas e enviadas à base de conhecimento relevante da estrutura de memória do agente.
- **Base de conhecimentos:** compreende os seguintes componentes:
 - *Objetos:* armazena a informação sobre os objetos no mundo do agente e o relacionamento entre eles;

- *Temporal*: representa a informação sobre a topologia do tempo. Por meio de axiomas, representa-se as características de tempo, restrições temporais sobre objetos, e as relações temporais entre agentes, objetos e o mundo;
- *Espacial*: similar à base de conhecimentos temporal, mas aplicada às informações espaciais;
- *Conhecimento especialista*: contém o conhecimento sobre um domínio específico, é o *know-how* do agente;
- *Conhecimento sobre outros agentes*: contém informações tais como o modelo dos outros agentes que o agente em questão necessita conhecer;
- *Conhecimento sobre o mundo*: descreve as informações de senso comum;
- **Recursos**: compreende os recursos disponíveis ao agente, incluindo os recursos computacionais;
- **Ações**: contém informações sobre o conjunto de operações primitivas que o agente é capaz de executar;
- **Crenças, compromissos e objetivos**: contém modalidades primitivas das crenças, compromissos e objetivos que descrevem o estado mental do agente. Estas primitivas são a base através da qual outros estados mentais de agentes são definidos, incluindo crenças e compromissos mútuos;
- **Planos**: especifica os métodos de planejamento disponíveis ao agente;
- **Aprendizagem**: contém algoritmos pelos quais um agente pode atualizar sua base de conhecimento, tais como indução, explanação, etc.;
- **Controle**: este controle de alto nível pode ser feito através de um meta-sistema e contém informações sobre o que o agente sabe, controle do espaço de busca e como aplicar restrições de integridade que assegure a consistência da base de conhecimentos.
- **Raciocínio**: é um componente complexo constituído por vários componentes que permitem ao agente raciocinar sobre diversos níveis hierárquicos. O mecanismo de raciocínio deve normalmente incluir, em adição ao raciocínio de lógica clássica, o raciocínio não-monotônico, raciocínio sobre crenças, tempo, espaço e ações.

O modelo formal apresentado por [TOR 95] é baseado em três modalidades primitivas de crenças, objetivos e comprometimentos. Modalidades mais complexas são definidas em termos destas modalidades primitivas. Duas espécies de agentes são consideradas: agentes simples e grupos de agentes. Um grupo de agentes é uma entidade abstrata que denota não apenas agentes individuais, como também subgrupos de agentes. Visto pelo lado de fora, um grupo de agentes pode ser entendido como um único agente. É utilizada a lógica de primeira ordem padrão, lógica modal e lógica temporal para descrever situações e lógica dinâmica para descrever planejamento.

A semântica do modelo é uma extensão da interpretação de Kripke de mundos possíveis [COS 92], onde cada mundo possível é uma estrutura temporal. Um plano conjunto de um grupo de agentes envolve um comprometimento no sentido de que cada um, entre todos os membros do grupo atinja satisfatoriamente seu compromisso. Este comprometimento conjunto é formalizado como uma restrição sobre as ações e as crenças. Um plano conjunto, por sua vez, é um conjunto de ações integradas que freqüentemente envolve condições estritas de sincronização.

2.2.1 Coordenação e comunicação em sistemas multiagentes

Serão apresentadas, a seguir, algumas considerações sobre a questão da coordenação e compartilhamento de conhecimentos entre agentes com a finalidade de determinar quais as alternativas foram consideradas mais adequadas a este projeto.

Um sistema multiagentes pode ser visto como uma rede, fracamente acoplada, de solucionadores de problemas que trabalham em conjunto para resolver problemas que vão além da sua capacidade individual. Estes solucionadores de problemas são essencialmente autônomos, distribuídos e, muitas vezes, heterogêneos em sua natureza [BAK 97].

É de vital importância a coordenação do comportamento dos agentes e da maneira pela qual eles compartilham seus conhecimentos, objetivos, habilidades e seus planos para, em conjunto, tomar as ações necessárias para solucionar um problema. Para que diferentes agentes autônomos possam cooperar mutuamente a fim de atingirem seus objetivos é necessário que a sociedade possua organização e comunicação. A organização diz respeito à natureza e à função da sociedade e de seus elementos constituintes e a comunicação é o principal instrumento que os agentes utilizam para desenvolver a coordenação de suas ações. Segundo [HÜB 94], estas duas necessidades colocam aos projetos de ambientes multiagentes os seguintes problemas:

- Como formular, descrever, decompor e alocar problemas entre um grupo de agentes inteligentes;
- Como habilitar agentes para comunicação e interação e qual a linguagem de comunicação e protocolo empregar;
- Como verificar que agentes agem coerentemente em suas decisões e suas ações;
- Como habilitar os agentes a representar e raciocinar sobre as ações, planos e conhecimentos a respeito dos demais agentes;
- Como reconhecer e tratar pontos de vistas conflitantes entre diferentes agentes ou grupos de agentes e coordenar as suas ações;
- Como projetar e implementar sistemas práticos de IAD.

[COS 94] analisa os problemas relacionados com a coordenação de sistemas multiagentes, com ênfase na possibilidade de migração de agentes entre diferentes sociedades. Segundo ele, quando um agente ingressa em uma sociedade, dois tipos de problemas são identificados para que o novo agente possa se adaptar a esta sociedade: o problema da *linguagem e interação* e o problema do *conhecimento e atuação*. O primeiro, refere-se ao uso das expressões adequadas e a maneira como as interações são organizadas na sociedade, enquanto que o segundo refere-se a que papel o agente deve desempenhar e que papéis ele deve esperar que demais agentes integrantes da sociedade desempenhem.

O quadro da figura 4, a seguir, adaptado de [BRE 98], resume a questão da cooperação entre agentes em sistemas multiagente, dividindo o sistema em níveis ou camadas que vão desde o problema da comunicação entre os agentes até as estratégias para a solução de problemas de forma cooperativa. A comunicação forma a base da cooperação e é constituída pelo transporte de mensagens entre os agentes através da rede, pelos protocolos de comunicação e pelos métodos de comunicação resultantes.

Os protocolos e as estratégias de cooperação, por sua vez, são construídos sobre os métodos de comunicação e são de fundamental importância na definição de: quem, quando e como devem atuar os agentes na solução do problema.

Cooperação	Estratégias			
	Protocolos			
Comunicação	Linguagens de comunicação			
	Quadro de avisos	Difusão de mensagens	Federação	Mensagens diretas
	Transporte			

FIGURA 4 – Comunicação e Cooperação em Sistemas Multiagentes.

Se diversos agentes trabalham de forma cooperativa na solução de um problema, em uma sociedade de agentes, é necessário definir uma estratégia a ser utilizada no processo de solução do problema. [WAL 98] divide a definição da estratégia em três etapas: a *divisão do problema em subproblemas*; a *solução dos subproblemas*; e a *combinação das soluções*. A figura 5 ilustra este conceito.

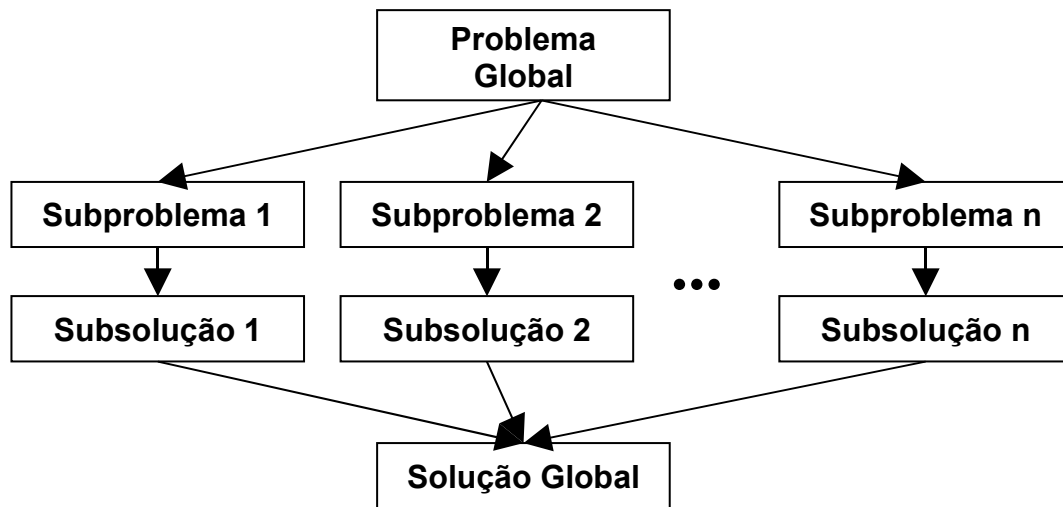


FIGURA 5 –Distribuição da Resolução de um Problema.

A divisão do problema em subproblemas e a designação de cada subproblema ao agente associado é uma etapa fundamental do processo. Isso normalmente é feito na etapa de análise e projeto do sistema. Mas, a fim de prover maior poder de atuação e maior grau de inteligência ao sistema, algumas decisões devem ser tomadas em tempo de execução, pelos próprios agentes. Para isso são definidos protocolos e estratégias de cooperação. Quanto mais autonomia e flexibilidade o sistema possuir, para definir a estratégia de solução do problema em tempo de execução, maior grau de adaptabilidade o sistema pode ter.

Como regra, pode-se estabelecer que, quanto mais fraca for a definição prévia do problema, como é o caso do processo de ensino em ambientes educacionais, maior é o grau de autonomia que o sistema deve possuir para definir, em tempo de execução, a melhor estratégia a ser adotada.

[DOR 97]apresenta uma definição das possíveis topologias de cooperação entre agentes, ilustrado na figura 6:

Se os agentes do sistema trabalham de forma completamente independente e seguem seus próprios objetivos, eles se classificam entre os sistemas *independentes*. Se, por sua vez, os agentes têm objetivos não-correlatos, eles podem ser considerados como *discretos*, caso contrário eles classificam-se como de *cooperação emergente*.

Sistemas são *cooperativos* quando possuem mecanismos explícitos de cooperação. Esta categoria de sistemas multiagentes pode ser considerada como *comunicativos* quando possuem e utilizam protocolos e procedimentos de cooperação. Caso contrário, se utilizam meios indiretos, são considerados *não-comunicativos*.

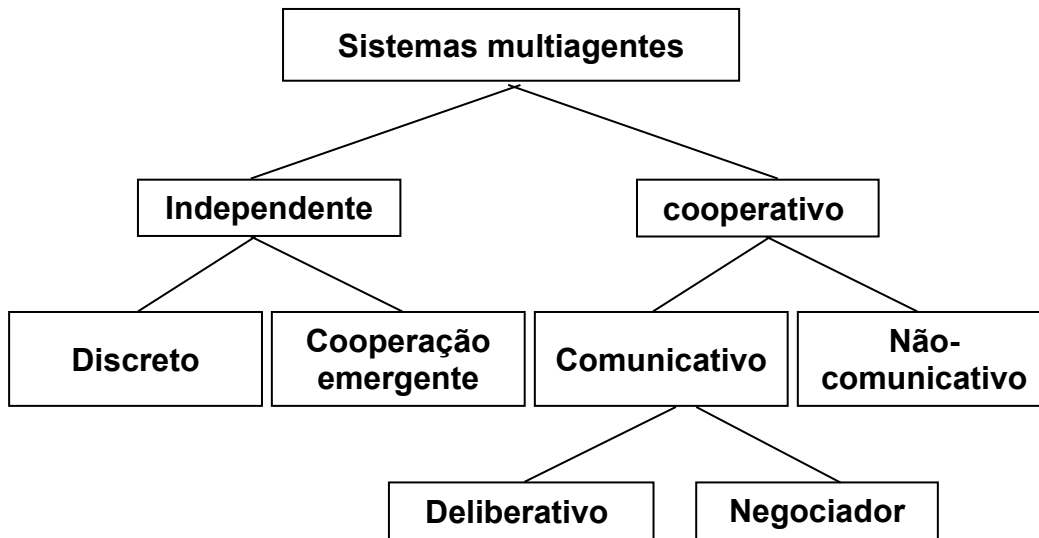


FIGURA 6 – Topologia de Cooperação Entre Agentes.

Sistemas multiagentes comunicativos podem ainda ser subdivididos em deliberativos e negociadores. Em sistemas *deliberativos*, ocorre um planejamento e um acordo a respeito da metodologia entre os agentes, enquanto que em sistemas *negociadores* existe um componente competitivo entre os agentes e uma metodologia de negociação deve ser utilizada para resolver conflitos e designar tarefas entre os agentes.

Nos sistemas baseados em negociação, dois ou mais agentes podem executar uma mesma tarefa ou solucionar um mesmo tipo de problema. Para tanto, é estabelecido um protocolo de negociação conhecido como rede de contratos (*contract net system*) [WAN 97]. O processo ocorre da seguinte forma: depois da fase da subdivisão do problema, um agente coordenador faz uma espécie de edital ou solicitação de ofertas. Os agentes, ao perceberem o edital avaliam se podem solucionar o problema, de acordo com as suas aptidões, e respondem ao coordenador, usualmente informando o custo da solução, relativo ao tempo ou carga de processamento. De posse dessas informações, o coordenador avalia e define um ou mais agentes para executar cada um dos processos para a solução de cada subproblema tipo de problema tratado por sistemas de ensino, no nosso ponto de vista, parece, no entanto, adequar-se melhor às características dos sistemas deliberativos.

Nesta proposta, assume-se como conceito de coordenação a propriedade de interação entre um conjunto de agentes na execução de alguma ação coletiva. Sem coordenação, os benefícios de solucionar um problema de forma descentralizada podem resultar, algumas vezes, em soluções caóticas, incoerentes e individuais. Uma coordenação adequada pode resultar em uma solução mais coerente, eficaz e efetiva ao problema [BAK 97].

Para que os agentes de um sistema possam cooperar entre si e coordenar as suas ações, é necessário que eles tenham conhecimento das capacidades, habilidades, desejos e planos dos outros agentes. Parte deste conhecimento pode ser obtida dos papéis que cada um dos agentes pode assumir em uma sociedade. O problema é como os agentes de uma sociedade percebem os papéis uns dos outros.

Para tal, os objetivos de um processo de coordenação devem assegurar que todas as partes constituintes de um problema sejam designadas a pelo menos um agente. As principais razões pelas quais as ações de múltiplos agentes devem ser coordenadas são:

- Existem dependências entre as ações dos agentes;
- Existem restrições globais do problema que devem ser consideradas;
- Nenhum agente possui competência, recursos e informações suficientes para solucionar inteiramente o problema.

Norman [DOR 97] propõe uma definição mais restrita de cooperação. Segundo ele, compartilhar de um mesmo propósito (implícito ou explícito), não garante, por si só, que agentes cooperem entre si. Dois operários pintando uma casa, por exemplo, estão imbuídos de um mesmo propósito, mas, se eles não souberem da existência um do outro, não haverá cooperação. Para que haja cooperação, é necessário uma ação conjunta, um plano de ação acordado entre os agentes. Isto não seria possível sem uma representação interna dos estados do mundo por parte do agente.

Segundo [HÜB 94], três diferentes mecanismos para identificar os papéis dos demais agentes de uma sociedade podem ser empregados:

- Por meio de um protocolo de apresentação: neste mecanismo, o agente interessado argúi os outros agentes. Para tal, é necessário que todos os agentes tenham conhecimento do mesmo protocolo de apresentação, tenham descrições comuns para papéis e tenham conhecimento dos seus próprios papéis.
- Pelo reconhecimento do processo que o agente executa: por este mecanismo o agente observador verifica uma seqüência de eventos globais em que o agente observado está participando e traça o comportamento deste agente.
- Pelo reconhecimento do plano do agente: por esta técnica o agente observador identifica a intenção do agente observado pelo reconhecimento do plano que está sendo executado pelo agente observado.

A arquitetura utilizada neste trabalho emprega um protocolo de apresentação executado durante a inicialização dos agentes.

Três níveis devem ser considerados na comunicação entre agentes, conforme foi mostrado anteriormente, na figura 4: o nível de transporte de mensagens, que trata da forma pela qual as mensagens são fisicamente transportadas entre os agentes; o nível de mecanismos de troca de mensagens; e o nível de linguagem de comunicação, que trata da forma como as mensagens são constituídas.

No nível de linguagem de comunicação, um trabalho muito consistente vem sendo desenvolvido pela ARPA (*Advanced Research and Projects Agency*), que patrocina um projeto denominado KSE (*Knowledge Sharing Effort*) que tem por objetivo o desenvolvimento de metodologias e software para o compartilhamento e a reutilização de conhecimento [ARP 98]. E que está se tornando um padrão de fato na implementação de sistemas multiagentes [BAK 97], [WAL 98].

Neste contexto, um agente é visto como um componente de software que se comunica com outros agentes através de mensagens codificadas em uma Linguagem de Comunicação de Agentes (*Agent Communication Language - ACL*). Os grupos de trabalho do projeto KSE desenvolveram uma ACL constituída por três camadas: uma *Camada de Protocolo*, que define a sintaxe das mensagens e os métodos de troca de informações; uma *Camada de Conteúdo*, que define a informação, propriamente dita, que é trocada; e uma *Camada Ontológica*, que provê uma base semântica específica para o domínio das expressões contidas na linguagem.

Camada de protocolo: o grupo KSE desenvolveu uma linguagem denominada KQML (*Knowledge Query and Manipulation Language*) para ser utilizada na camada de protocolo da ACL. A especificação da linguagem KQML [FIN 92] define o formato das mensagens e um conjunto de elementos denominados *performatives*, que são independentes do domínio, correspondendo a categorias de mensagens semanticamente distintas. Cada mensagem KQML deve obrigatoriamente conter uma *performative* (por exemplo: *tell, ask-if, reply, achieve, etc.*) e campos que incluem a identificação do destinatário, da origem, e outros. A especificação da linguagem KQML assume que a comunicação entre os agentes é confiável, e que a transferência de mensagens ocorre de um modo assíncrono em relação à operação dos agentes. Além disso, ela é neutra em relação ao método usado para codificar a mensagem, que pode ser desde uma cadeia arbitrária de caracteres até uma expressão SQL de consulta a um banco de dados.

Camada de conteúdo: o grupo KSE desenvolveu também uma proposta denominada KIF (*Knowledge Interchange Format*) com o objetivo de codificar o conteúdo das mensagens KQML. A linguagem KIF [GEN 95] define uma sintaxe, uma semântica e um conjunto de operadores, baseados em lógica de predicados de primeira ordem com extensões para suportar raciocínio meta-nível e inferências não-monotônicas.

Ontologias: uma vez que a linguagem KIF não define os termos, relações, funções, etc., específicos para modelar algum domínio em particular, necessário para a comunicação entre sistemas arbitrários baseados em arquiteturas multiagentes, estes elementos devem constituir uma ontologia. No contexto do compartilhamento do conhecimento, uma ontologia é uma especificação de uma conceituação, isto é, uma descrição dos conceitos e relacionamentos que podem existir em uma comunidade de agentes. Ontologias reutilizáveis são importantes para a integração de informações, desenvolvimento de bases de dados compartilhadas e interoperação de sistemas [FIN 98].

No nível referente aos mecanismos de comunicação, é possível identificar quatro diferentes topologias de trocas de mensagens entre agentes: a topologia de *mensagens diretas*, de *federação*, de *difusão de mensagens* e de *quadro de avisos* [BAK 97][WAL 98].

O modelo de **troca direta de mensagens**, ou comunicação direta envolve o estabelecimento de uma ligação direta entre os agentes, através de um conjunto de protocolos que permite o estabelecimento de comunicação fim a fim. Este tipo de comunicação implica que cada um dos agentes envolvidos deve ter conhecimento da existência dos seus parceiros e da forma de endereçar mensagens a eles

O modelo de **sistemas federados** deve ser utilizado quando o número de agentes é muito grande. Neste modelo, uma estrutura hierárquica de agentes é estabelecida e a troca de mensagens dá-se através de mediadores ou facilitadores.

O modelo de **comunicação por difusão de mensagens** (*broadcasting*) é utilizado quando for necessário alguma comunicação a todos os agentes do sistema, ou quando o remetente da mensagem não sabe quem é, ou qual o endereço do seu destinatário.

No modelo de comunicação, através de **quadro de avisos** (*blackboard*), a comunicação dá-se através de algum tipo de compartilhamento de memória utilizado como um repositório no qual os agentes obtêm informações e postam suas mensagens a

outros agentes. Usualmente, o quadro de mensagens é particionado em vários níveis de abstração, apropriado a cada problema em particular.

A arquitetura do JADE utiliza os mecanismos de troca direta de mensagens, por meio da camada de transporte e o sistema de quadro de avisos, através do compartilhamento de uma base de dados.

Quanto ao nível da camada de transporte, encarregada de rotear as mensagens entre os agentes, são utilizados o conjunto de protocolos e serviços disponíveis na Internet para comunicação direta entre agentes, através do estabelecimento de *sockets* de comunicação entre eles, e de *multicast sockets*. Avanços mais recentes nas linguagens de programação, entretanto permitem formas mais eficientes e sofisticadas de comunicação.

Durante as diversas fases de evolução do JADE foram utilizadas, além de *sockets* TCP/IP os mecanismos RMI da linguagem JAVA e RPC da linguagem XML.

2.2.2 Agentes pedagógicos

Um dos maiores problemas com os sistemas tradicionais de aprendizagem auxiliados por computador é a sua dificuldade em prover um ensino individualizado, adaptado ao aluno. Os sistemas tutoriais inteligentes (ITS ou ICAI) buscam um sistema de ensino mais eficaz através do diagnóstico cognitivo do aluno. A modelagem do aluno permite a realimentação adequada, ao sistema, por meio da representação dinâmica do estado cognitivo de cada aluno, individualmente, determinando o que ensinar e como ensinar, em cada momento.

Segundo [WEB 97], as principais funcionalidades dos ITS são o controle da seqüência curricular, através da orientação ao aluno, de forma adaptativa, referente às atividades a serem desenvolvidas, e o suporte interativo à solução de problemas, através de mecanismos de ajuda individualizada para as características e para as habilidades que o aluno possui num dado momento. Essas funcionalidades diferenciam os sistemas inteligentes dos sistemas tradicionais de ensino por computador, na medida em que eles incorporam técnicas inteligentes que utilizam a metáfora de reproduzir as habilidades que os professores usam em sala de aula ou no ensino individual.

A maioria dos ITS utilizam como técnica para manter o modelo do aluno atualizado, o diagnóstico da solução de exercícios feitos pelo aluno. Alguns, no entanto, procuraram utilizar formas mais avançadas de modelagem monitorando todos os estágios de solução dos problemas feitos pelo aluno. A arquitetura do JADE busca alcançar esta solução, monitorando várias ações executadas pelo aluno a fim de obter um diagnóstico, não só da performance do aluno, como também de seu perfil.

Em sua forma tradicional, nos ITS, o aluno é modelado em termos de domínio de conhecimento da matéria, objeto de estudo, normalmente registrando-se as unidades de conhecimento que o sistema conhece e o estudante não (modelo de *overlay*), ou ainda, adicionalmente, as unidades de conhecimento que o aluno conhece e o sistema não (modelo de erros). Dentro de uma abordagem de sistemas especialistas, em ambos os casos, o modelo de aluno é utilizado por uma máquina de inferência para desenvolver a atividade de ensino-aprendizagem.

Segundo [MAS 96], embora a abordagem de ITS tenha seus méritos, têm sido detectados muitos problemas associados a ela. Na abordagem descrita acima, o estudante é modelado em termos do modelo de representação do sistema especialista. Entretanto, nem sempre é tão óbvio que este modelo de representação corresponda ao

modelo de representação cognitiva do aluno. Como a performance do aluno é interpretada em termos do modelo de representação do sistema especialista, o grau de sensibilidade de um ITS é determinado pela extensão do conjunto de comportamentos possíveis do aluno que preenchem a estrutura de representação do ITS.

Os ITS, em sua maioria, foram projetados para serem utilizados em sala de aula, geralmente acompanhados de um professor ou monitor que eventualmente possa ajudar o aluno quando o auxílio propiciado pelo sistema falhar ou for insuficiente. Já, um sistema voltado para Educação a Distância deve cobrir ao máximo a dificuldade causada pela distância física que separa o aluno do professor e dos demais colegas, exigindo assim mecanismos mais eficientes de adaptabilidade e, principalmente, de ajuda ao aluno na solução de problemas.

Para isso, o sistema deve desempenhar o papel do professor tanto quanto possível, construindo um modelo de aluno robusto, para cada usuário, em particular, que permita de forma individual e adaptativa:

- adaptar o currículo a cada usuário;
- ajudá-lo a navegar nas atividades do curso;
- dar suporte ao trabalho a ser realizado ou aos exercícios e problemas a serem resolvidos;
- prover recursos de ajuda sempre que necessário.

Como a performance do aluno raramente é consistente, e é impossível prever toda a gama de comportamento do aluno, a adaptabilidade dos ITS é limitada, e os modelos de ITS clássicos não são robustos suficientes para prover os requerimentos mínimos necessários para um ambiente interativo de aprendizagem que, segundo [MAS 96], devem ser os seguintes:

- *Interatividade*: a interação em ambientes de aprendizagem pode ser considerada como um diálogo cooperativo entre dois parceiros, no qual algumas regras especiais são aplicadas para promover a aprendizagem;
- *Instrução adaptável*: em um sistema instrucional, deve haver uma adaptação, momento a momento, às necessidades individuais do aluno;
- *Robustez*: o sistema interativo deve atender em tempo hábil às mudanças de comportamento do aluno, além de erros e comportamentos inesperados;
- *Monitoramento direto do processo de aprendizagem*: um ILE deve suportar a otimização do processo de aprendizagem do aluno;
- *Avaliação empírica*: todos os estágios do processo de desenvolvimento de sistemas interativos de aprendizagem devem ser baseados em pesquisa fundamental sobre o processo de aprendizagem que ele suporta e em teste contínuo de cada fase de desenvolvimento dos protótipos;
- *Parcimônia*: uma arquitetura de um sistema interativo de aprendizagem deve ser simples e eficiente.

Os avanços mais recentes no campo dos ambientes de aprendizagem inteligentes têm proposto o uso de arquiteturas baseadas em sociedades de agentes. Os princípios

dos sistemas multiagentes têm mostrado um potencial bastante adequado ao desenvolvimento de sistemas de ensino, devido ao fato de a natureza do problema de ensino-aprendizagem ser mais facilmente resolvido de forma cooperativa. Por isso, o JADE foi desenhado sob este paradigma.

Segundo [OLI 96], a abordagem de aplicação de técnicas de DAI em ambientes de ensino-aprendizagem está diretamente vinculada ao paradigma adotado no que concerne à teoria de ensino-aprendizagem. Quando o ensino é visto como uma tarefa, um problema a ser resolvido de forma cooperativa entre todos os agentes envolvidos no processo, um objetivo a ser atingido, um conjunto de estratégias e planos devem ser levados a cabo. A tarefa de ensino é reduzida ao problema de geração de um plano para atingir os objetivos definidos pelo projetista. Nesta abordagem, a principal contribuição de DAI é a possibilidade de melhorar a performance do sistema através da divisão do problema em subproblemas com algum grau de processamento paralelo.

Esta abordagem é chamada de *DAI based learning environments*. As arquiteturas desenvolvidas nesta abordagem, usualmente, são variações das arquiteturas funcionais dos ITS's, onde cada função é implementada por um ou mais agentes especializados. Como normalmente ocorre em DPS, cada agente tem uma função fixa e bem definida e o plano geral de ensino é gerado pela combinação das contribuições de diferentes agentes, embora, para o estudante, o sistema é visto como um único indivíduo, apesar de que, internamente seja constituído por uma sociedade.

Quando o processo de ensino-aprendizagem é visto sob uma abordagem cooperativa e centrada no aluno, como se pretende neste projeto, um ambiente desenvolvido sob a perspectiva de uma sociedade constituída por vários agentes autônomos (humanos e/ou artificiais) é mais adequada [OLI 96]. Numa sociedade assim constituída, alguns agentes atuam como tutores e outros como alunos, com possibilidade de eventualmente trocarem de papéis. Todos os agentes são envolvidos na construção de um corpo comum de conhecimentos em um domínio particular. Cada agente, por sua vez, pode ou não ser dividido em agentes internos (subagentes ou agentes locais), podendo constituir-se numa sociedade interna e numa sociedade externa.

Dessas idéias surge o conceito de Agentes Pedagógicos que se constituem em agentes orientados às atividades de ensino. [GIR 98]propõe uma taxinomia de agentes pedagógicos na qual divide as arquiteturas de ambientes de aprendizagem em *sistemas orientados a objetivos*, divididos em tutores, mentores e assistentes e *sistemas orientados à utilidade*, tais como MOO e Agentes de WWW.

As tabelas, a seguir, mostram as características dos agentes pedagógicos orientados para objetivos, de acordo com a taxinomia proposta:

TABELA 2 - Taxinomia de Agentes Pedagógicos

	Tutor	Mentor	Assistente
Conhecimentos acerca do ambiente	Forte	Forte	Médio
Conhecimento do domínio	Forte	Forte	Forte
Modelo do aluno	Forte	Médio	Fraco ou nulo
Aspectos pedagógicos	Forte	Médio	Fraco ou nulo

TABELA 3 - Classificação das arquiteturas

Propósito Educacional	AMBIENTES:		
Propriedades dos agentes:	Tutor	Mentor	Assistente
Autonomia	1,2,5,6,8,10,11,13,14,15	3, 7	4, 9
Habilidade Social	1,2,5,6,8,10,11,12,13,14,15	3, 7	4, 9
Comportamento Pro-ativo	1,2,3,4,5,6,8,10,11,12,13,14,15	3, 7	9
Comportamento Reativo	1,2,5,6,7,8,10,14	3	4, 9
Persistência ou Continuidade	1,5,6,8,10,11,12,13,14,15	3, 7	4, 9
Temporalidade	2		
Personalidade	1,7,5,6,10,12,14,15		9
Aprendizagem	5		
Flexibilidade	5, 10		

TABELA 4 - Referências Bibliográficas sobre ILE Apontadas na TABELA 3

Nº	REFERÊNCIA	DESCRIÇÃO
1	[AND 97]	Persona - ITS para treinamento sobre utilização de modem.
2	[COS 97]	Conjunto de ITS para ensinar conteúdos específicos.
3	[COO 97]	ITS para ensino de conceitos de Música.
4	[DIL 97]	ITS de ajuda para estudantes.
5	[FRA 97]	ITS para ensinar conceitos de radiologia em Medicina.
6	[LEW 97]	ADELE - tutor pessoal para cada aluno na WEB.
7	[LES 97]	ITS para ensino de Biologia para crianças.
8	[MOR 97]	ITS para ensino de conceitos de Matemática.
9	[MUR 97]	ITS para selecionar preferências de estudantes.
10	[RIC 97]	STEVE - ITS para ensinar alguns recursos de laboratório.
11	[RIT 97]	ITS para ensino de conceitos de Matemática.
12	[HIE 97]	EduAgents - Agentes que auxiliam estudantes a resolver problemas de equações elementares.
13	[LER 96]	Ambiente cooperativo com agentes para auxiliar estudantes a aprender sobre tecnologia.
14	[GIR 97]	Multi-Ecological - ILE para ensinar conceitos sobre poluição.
15	[SIL 97]	Eletrotutor - ITS para ensinar conceitos de eletricidade.

Na taxinomia de [GIR 98], o JADE é enquadrado, na TABELA 3, através da análise do protótipo Eletrotutor II [SIL 97](item 15, na TABELA 4) na categoria de Tutor, segundo os critérios da TABELA 2. Entretanto, com os avanços obtidos posteriormente e a elaboração dos novos protótipos o JADE contempla também os fatores denominados *comportamento reativo* e *flexibilidade* referidos na TABELA 3.

2.3 Projeto e implementação de Sistemas Multiagentes

Nos últimos anos, algumas ferramentas para a criação de sistemas multiagentes têm sido desenvolvidas. Estas ferramentas vão desde plataformas e *frameworks* para desenvolvimento de sistemas multiagentes a linguagens de programação para a elaboração de programas orientados ao paradigma de agentes, como também a linguagens de comunicação nas quais agentes são descritos ou especificados.

[BAK 97], cita diversas ferramentas deste tipo e faz uma comparação entre algumas delas, além de apresentar a ferramenta JAFMAS. As ferramentas citadas por este autor são:

IBM Aglets Wokbench: um ambiente visual para a construção de aplicações em linguagem Java, baseadas em rede, que utilizam o conceito de agentes móveis para funções de busca, acesso e manuseio de dados corporativos e outras informações [IBM 99].

Concórdia: infra-estrutura (*framework*) baseada em Java desenvolvida pela *Mitsubishi Electric information Technology Center of America* para o desenvolvimento de agentes móveis. [MIT 99].

Odissey: é uma implementação em Java da ferramenta *General Magic* concebida em linguagem *Telescript* desenvolvida para aplicações de agentes móveis [GEN 99].

Voyager: desenvolvida pela *Objectspace Inc*, permite programadores Java criar sofisticadas aplicações utilizando técnicas tradicionais e orientadas a agentes na aplicação de sistemas distribuídos, incluindo agentes móveis [OBJ 99].

JATLite: desenvolvida pelo *Computer Science department at Stanford University*, provê um conjunto de *Java Packages* que constitui uma infra-estrutura para o desenvolvimento de sistemas multiagentes em linguagem Java. Provê uma estrutura básica de comunicação que permite a troca de mensagens KQML entre agentes [STA 99].

InteRRaP: desenvolvida no *German Artificial Intelligence Research Institute (DFKI)* para a elaboração de projetos de sistemas de agentes híbridos (reativos e cognitivos) destinados às aplicações de robótica e projetos industriais. Apesar de eficiente, apresenta a desvantagem de ser desenvolvida em linguagem pouco conhecida e especificamente para plataforma Unix [GER 99].

DMARS: desenvolvido em linguagem C++ pelo *Australian Artificial Intelligent Institute* como um ambiente de implementação e desenvolvimento orientado a agentes de aplicações com sistemas multiagentes de acordo com o modelo BDI (*beliefs, desires and intentions*) [AUS 99].

Agent Talk: é uma linguagem de descrição de protocolo de coordenação de sistemas multiagentes desenvolvida pela *NTT Communication Science Laboratories, Ishida laboratory, e Kyoto University*. Trata-se de uma linguagem de programação capaz de implementar protocolos e agentes que atuam de acordo com o protocolo criado. Provê uma máquina de estados finitos estendida para a representação explícita destes protocolos. É implementado em linguagem *Common Lisp* para ambiente Unix [NTT 99].

Telescript: desenvolvida pela *General Magic* foi a primeira implementação comercial do conceito de agentes móveis. [GEN 99a].

AgentTCL: é também uma linguagem para implementação similar à *Telescript*, desenvolvida na *Dartmouth University*. Baseada na linguagem de *TCL script* é bastante simples e versátil. Necessita de um interpretador executando os agentes e um servidor em cada máquina. A comunicação ocorre através de *sockets* TCP/IP e possui inúmeras facilidades para troca de mensagens, criação de novos agentes e clonagem de agentes existentes e mecanismos de segurança. [DAR 99].

Swarm: desenvolvido como uma plataforma de software pelo *Santafe Institute* para a simulação de sistemas adaptativos complexos. Utiliza *GNU Objective C Compiler* e executa em ambiente Unix e X Windows. É uma ferramenta útil para pesquisas na área de Vida Artificial [SAN 99].

Echelon: é um produto comercial que provê uma arquitetura completa de hardware e software para sistemas de sensores, monitores e controle [ECH 99].

Agent Builder: é um conjunto integrado completo, desenvolvido pela *Reticular Systems* para construção de sistemas multiagentes, em Java, constituída por dois componentes principais: um kit de ferramentas e um ambiente de execução (*run time*). É baseado na construção de agentes com arquitetura BDI proposta por [SHO 93]e suporta a comunicação entre agentes através de linguagem KQML. [RET 99].

JAFMAS: é uma tese de doutorado que propõe uma metodologia e provê uma estrutura para desenvolvimento de sistemas multiagentes, oferecendo uma biblioteca de classes em Java que permite a implementação de sistemas com ênfase na comunicação entre agentes através da linguagem KQML [DEE 98].

A TABELA 5, a seguir, adaptada de [BAK 98] faz um quadro comparativo das ferramentas orientadas ao desenvolvimento de sistemas multiagentes em linguagem JAVA.

TABELA 5 – Comparação entre Ferramentas de Desenvolvimento.

	IBM Aglets	Odyssey	Voyager	JAFMAS	Agent Builder	JATLite
Versão Java	JDK 1.1	JDK 1.1	JDK 1.1	JDK 1.1	JDK 1.1	JDK 1.1
Cada agente tem sua <i>thread</i>	Sim	Sim	Sim	Sim	Sim	Sim
Manejo dos agentes	Objeto externo	Objeto externo	Objeto externo	Java virtual machine	Objeto externo	Objeto externo
Formas de comunicação	Direta	Direta	Direta	Direta e multicast	Direta	Direta
Nº de pontos de acesso ao agente	Um	Um	Múltiplos	Um		
Forma de identificação dos agentes	Agente externo	Agente externo	Agente externo	multicast		Agente externo
Suporte a KQML	Não	Não	Não	Sim	Sim	sim
Provê meios de coordenação	Não	Não	Não	Sim		
Mobilidade dos agentes	Sim	Moderado	Sim	Não	Não	Não
Facilidades de segurança	Sim	Utiliza as facilidades do Java	Utiliza as facilidades do Java	Utiliza as facilidades do Java		
Encapsulamento de dados	Moderado	Moderado	Moderado	Completo		
Interface visual	Muito realçada	Moderada	Moderada	Apenas para monitoração e criação de agentes		

2.4 Trabalhos correlatos

Nos últimos tempos tem sido considerável o número de trabalhos que vêm sendo desenvolvidos na área de Informática Aplicada à Educação. Em trabalhos anteriores [SIL 96a] e [SIL 98] foram discutidos diversos aspectos da evolução do emprego e das diversas abordagens da Informática na Educação.

De um modo amplo, o ensino baseado em computador pode ser classificado, de acordo com Barr & Feigenbaum [BAR 82], Lawler & Yazdani [LAW 87], Vicari [VIC 89], Clancey & Soloway [CLA 90] e Giraffa [GIR 91], em cinco abordagens principais:

- CAI – *Computer Assisted Instruction* (Instrução assistida por computador - IAC);
- Programação em LOGO;
- Ambientes de demonstração, simulação e jogos;
- ICAI – *Intelligent computer assisted instruction* ou ITS – *Intelligent Tutoring Systems*;
- ILE – *Intelligent Learning Environment* (Ambiente Inteligente de Aprendizagem).

A tendência de avanço dos sistemas tutoriais inteligentes em direção aos vários tipos de ambientes de aprendizagem tem tido razões tanto computacionais quanto educacionais, isto é, pode ser explicada em termos de uma nova compreensão da Psicologia da Aprendizagem, da reconhecida dificuldade da construção de sistemas tutoriais e do apelo da abordagem de Ensino Centrado no Aluno [SEL 92].

Conforme foi discutido anteriormente, os avanços mais recentes no campo dos ambientes de aprendizagem inteligentes, têm proposto o uso de arquiteturas baseadas em sociedades de agentes, as quais têm mostrado um potencial bastante adequado ao desenvolvimento de sistemas de ensino, devido ao fato de a natureza do problema de ensino-aprendizagem ser mais facilmente resolvido de forma cooperativa.

Serão abordados brevemente, a seguir, alguns trabalhos nesta área, os quais foram considerados no desenvolvimento desta tese, com interesse especial naqueles que utilizam técnica de Inteligência Artificial Distribuída, e os trabalhos correlatos desenvolvidos no Grupo de Inteligência Artificial em Educação do Programa de Pós-Graduação em Ciência da Computação da UFRGS.

O sistema GIA (*Generic Instructional Architecture*) [CHE 95] é uma infraestrutura baseada no modelo de agentes, desenvolvida com a finalidade de dar suporte a um rápido desenvolvimento de ambientes inteligentes de ensino. Sua abordagem principal é prover técnicas que encorajem a reutilização e o compartilhamento de códigos e estruturas de dados, desenvolvendo uma arquitetura e um conjunto modular de componentes de software que propiciem uma base para o desenvolvimento de ambientes de ensino.

A sociedade de agentes do APPEAL [MAS 96] foi estruturada de modo a associar um agente a cada um dos diferentes aspectos de comportamento de um professor decomposto em: *navegação, prática, instrução, realimentação, apresentação e modelagem do aluno*. Cada aspecto foi implementado como um agente, desenhado para desempenhar especificamente cada uma destas funções. O comportamento geral do sistema emerge da interação livre entre os agentes.

O sistema SEI [TED 97] é constituído pelos seguintes agentes: Agente Controlador, Agente Estudante, Agente Domínio, Agente Tutor e Agente Comunicador. Cada um dos agentes é constituído como um *Java thread*, que executa de maneira contínua, e se comunica através da troca de mensagens KQML. O usuário acessa o agente *Comunicador* (através da interface do sistema), utilizando qualquer *browser WWW* que esteja habilitado para Java.

O projeto CALAT [NAK 97][NAK 96] implementa um Sistema Tutor Inteligente baseado em WWW. O sistema CALAT (*Computer Aided Learning and Authoring environment for Tele-education*), baseia-se no tutor CARNEY [FUK 95]. Ambos fazem parte de um programa de pesquisa dos laboratórios da NTT, empresa japonesa de serviços de telecomunicações para o desenvolvimento de *coursewares* de treinamento no campo das telecomunicações, tais como, Comutação Digital, Telefonia Móvel, RDSI, etc.

A atual implementação tem uma estrutura cliente-servidor baseada na arquitetura WWW. o servidor consiste em um *demon WWW* convencional chamado HTTPD e um processo tutorial auxiliar denominado Processo ITS. Para cada aluno que se conecta ao sistema é invocado um Processo ITS. O usuário acessa ao sistema utilizando um *browser* convencional de propósito geral. Um mecanismo de identificação de usuário é introduzido para manter a correspondência entre o processo ITS e o aluno, através do protocolo HTTP. Quando o botão de comando do CALAT é inserido no ambiente do aluno, este mecanismo faz com que cada link URL representando um comando WWW, inclua a identificação do aluno.

Três diferentes tipos de páginas WEB constituem o material instrucional: explanação, exercícios e simulação que, de modo similar ao Eletrotutor [SIL 92], e que foram adotadas no JADE, é associada a um sub-objetivo e compõem as estratégias de ensino. As páginas de explanação podem ser qualquer tipo de página HTML, podendo incluir *Java applets* ou ainda componentes multimídia. Os exercícios são constituídos por formulários HTML gerados dinamicamente, podendo ter respostas do tipo Verdadeiro ou Falso, Descrição ou ainda Seleção. As páginas de simulação permitem ao aluno adquirir conhecimentos procederia através de sistemas simulados.

Para prover capacidade de adaptação individual dinamicamente, o processo ITS implementa um modelo de aluno através de uma estrutura de árvore representando os objetivos do curso e as páginas associadas a cada um destes objetivos. De acordo com o nível de compreensão do aluno e na medida que os objetivos vão sendo atingidos, o que é checado através dos exercícios e simulações, o processo ITS determina a seqüência das páginas a serem apresentadas ao aluno.

O sistema CALAT possui ainda um subsistema de autoria de cursos. Os mecanismos descritos acima têm permitido implementar com sucesso diversos cursos que já estão disponíveis na rede.

O projeto ELM-ART II [WEB 97] propõe um curso introdutório de linguagem LISP baseado em WWW como um *Intelligent Learning Environment* (ILE) que suporta o ensino de programação baseado em exemplos, além de análise de solução de problemas e facilidades de testes e depuração de programas aos alunos.

A implementação do ELM-ART II contemplou a transcrição de textos impressos, utilizados pelas versões anteriores do projeto, em hyper-textos eletrônicos na forma de arquivos HTML, dividindo o conteúdo em pequenas subseções com páginas de textos organizados em conceitos a serem ensinados ao aluno. Estes conceitos foram

relacionados entre si de acordo com uma estrutura de pré-requisitos e organizados em forma de uma rede conceitual a fim de constituir uma base de conhecimentos. Todas as interações feitas pelo aluno com o sistema são registrados em um modelo individual de aluno, no qual, a cada página visitada, a correspondente unidade na rede conceitual é assinalada.

Quando uma página de texto é apresentada ao aluno através do *browser*, os *links* relacionados àquela unidade ou seção, bem como uma visão geral do curso, referentes à situação do aluno individualmente, são mostrados utilizando-se uma metáfora de um sinal de trânsito (um semáforo), ou seja, uma bola vermelha em frente aos *links* indica que a seção correspondente ainda não deve ser visitada pelo aluno, de acordo num dado momento, devido à falta dos pré-requisitos necessários. Uma bola verde indica que a sessão ou unidade é recomendada ao aluno e uma bola amarela indica que o link pode ser visitado mas não é especialmente recomendado ao aluno naquele momento.

O ELM-ART II permite interatividade com o aluno através de exemplos e de ferramentas de diagnóstico para a solução de problemas. Quando o aluno aciona um *link* correspondente a um exemplo, a avaliação da função chamada é mostrada em uma janela de um avaliador similar à de um ambiente LISP. Além disso, os alunos podem digitar programas LISP como solução de problemas propostos e submetê-los ao servidor que avalia e diagnostica erros no programa.

O projeto ADE – *Advanced Distance Education* [JOH 97], tem por objetivo o desenvolvimento de cursos interativos, com resposta rápida, e que possuem capacidade de adaptação às respostas do aluno, de acordo com a sua performance individual, além de manter um registro de suas ações para referência futura.

Segundo [JOH 97], embora os cursos desenvolvidos para *Web* tenham inúmeras vantagens sobre materiais convencionais, como livros-texto e cadernos de notas, eles possuem inúmeras deficiências:

- O acesso ao material do curso costuma ser muito lento, cansando o aluno;
- o curso não se adapta automaticamente a cada estudante, individualmente;
- apenas certo grau de interatividade costuma ser implementado através de programação, usando CGI ou Java;
- características de processamento na *Web* como *caching* e ocultação de informação do lado cliente interferem na coleta de dados de performance do aluno.

O principal responsável pela performance dos cursos em ADE é a ação de um agente denominado Adele – *Agent for Distance Learning Environments*. Uma cópia de Adele é executada no ambiente do aluno e interage com o aluno à medida que ele trabalha com o material do curso. Este agente é responsável pela monitoração do aluno, pelo registro e gravação de suas ações, pela adaptação da apresentação do material do curso, sempre que necessário, pelo relato da performance do aluno ao servidor central, ao final da sessão. O suporte à autoria de curso é dado através da definição do material de curso e através da informação ao agente de conhecimentos da matéria, objeto do curso, requeridos para que ele possa dar assistência ao aluno.

O projeto ADE desenvolve cursos na área médica e utiliza uma metodologia denominada PBL *Problem based learning* na qual quatro elementos-chaves são fornecidos ao aluno: primeiramente, o aluno é submetido a uma narrativa instrucional que pode ser constituída por textos ou material audiovisual; a seguir, é apresentado ao aluno um caso em forma de problema interativo que ele deve solucionar; todo material

de referência necessário é fornecido ao aluno. Finalmente, o aluno participa de uma discussão em tempo real com colegas e instrutores, utilizando ferramentas de *chat* e teleconferência. Cada módulo é centrado em torno de um caso ou problema que o aluno deve resolver.

Em vez de utilizar processamento centralizado em um servidor central, ADE utiliza uma abordagem descentralizada na qual uma parte da gerência do curso é feita em um servidor central, mas o material do curso é utilizado localmente utilizando programas executados no ambiente do aluno. Toda a interatividade e a capacidade tutorial é implementada no lado cliente. Embora todo o software cliente seja desenvolvido para ser utilizado com um navegador convencional, com material de referência utilizando HTML e interatividade obtida através de programas em Java, o ambiente implementado utiliza a maior parte de material distribuído previamente por meio de CDROM.

O servidor central é constituído por um servidor *Web*, por *scripts* de manuseio e uma base de dados que registra as informações a respeito do trabalho de cada aluno. O servidor *Web* é utilizado basicamente como interface entre o aluno e os instrutores e a base de dados, com o objetivo de administrar o curso.

A função do agente é monitorar as ações do aluno sobre um paciente virtual. Para isso é implementado um modelo mental de paciente, constituído por um conjunto de atributos revelados através de ações que o aluno exerce sobre o paciente; um modelo de tarefas que descreve as ações executadas pelo aluno, e um conjunto de estratégias pedagógicas que o agente pode executar.

A forma como o ADE desenvolve os cursos faz com que os mesmos tenham uma arquitetura bastante dependente do conteúdo para o qual foram concebidos. Parece difícil de desenvolver novos cursos aproveitando o código implementado, ou mesmo atualizar o conteúdo existente de um curso já desenvolvido

O sistema MCOE – *Multi-agent Co-Operative Environment* [GIR 98a], desenvolvido no Grupo de Inteligência Artificial em Educação do Programa de Pós-Graduação em Ciência da Computação da UFRGS, é a implementação de uma arquitetura multiagentes, que combina agentes reativos e cognitivos, e que apresenta uma metáfora de um ecossistema com vários elementos (diversos tipos de peixes, plantas e microorganismos) e personagens (turista, governante, etc.) que podem atuar sobre o ecossistema de forma positiva ou negativa, sob o ponto de vista de poluição e controle do equilíbrio ambiental.

O objetivo do sistema é propiciar um ambiente onde o aluno possa exercitar a interação com um ecossistema e avaliar os resultados benéficos e maléficos das atitudes exercidas pelo personagem e, dessa forma, construir o seu próprio conceito de equilíbrio ecológico e a consciência de preservação ambiental.

A arquitetura do sistema é constituída por uma sociedade heterogênea formada por diversos agentes reativos, que são os elementos do cenário, que sofrem ações do meio ambiente, e por agentes cognitivos, que utilizam estratégias para agirem sobre o meio ambiente e, por isso, possuem um conhecimento do ambiente. A figura 7 mostra a distribuição dos agentes no ambiente.

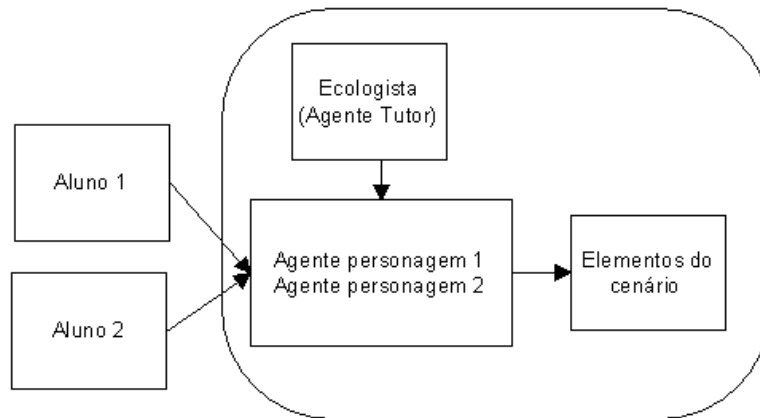


FIGURA 7 - A Distribuição de Agentes do MCOE

Os agentes do cenário são peixes, plantas, água, ecômetro, etc. Estes agentes têm um conjunto de comportamentos reativos, relacionados com o nível de poluição do ambiente. A poluição é diretamente associada ao nível de energia de cada elemento no cenário, e as suas reações são modeladas em função do impacto da poluição.

A interação do aluno ocorre através de uma atividade lúdica por meio de uma interface gráfica onde os elementos são inseridos. O aluno ou o professor definem previamente a configuração inicial do ambiente e o aluno visualiza então um lago em equilíbrio que começa a sofrer a ação dos elementos. O desafio do aluno é manter o equilíbrio ecológico e combater a ação dos elementos através das ferramentas que o personagem dispõe.

O sistema trabalha com dois tipos de agentes cognitivos: o Ecologista e o Aluno. O Ecologista não pode prever quais as estratégias que serão usadas, mas deve trabalhar em cooperação mútua com o aluno, observando as ações realizadas e seus resultados.

O agente Ecologista (tutor) possui uma arquitetura dotada de um conjunto de crenças acerca do ambiente, representado pela consequência das ações realizadas pelo Aluno ou pelo Ecologista, que utilizam ferramentas de seu personagem, apoiado em regras para ajudá-lo a decidir que ações deveria tomar e sensores para receber informações sobre o ambiente.

O modelo do aluno tem uma representação do estado mental através de um conjunto de crenças, desejos e expectativas sobre o ambiente, sobre si mesmo, e sobre o agente Ecologista. Possui também conhecimentos a respeito das ferramentas para manusear os elementos, sensores para receber informações ambientais e janelas de diálogo. A arquitetura do agente tutor é mostrada na figura 8.

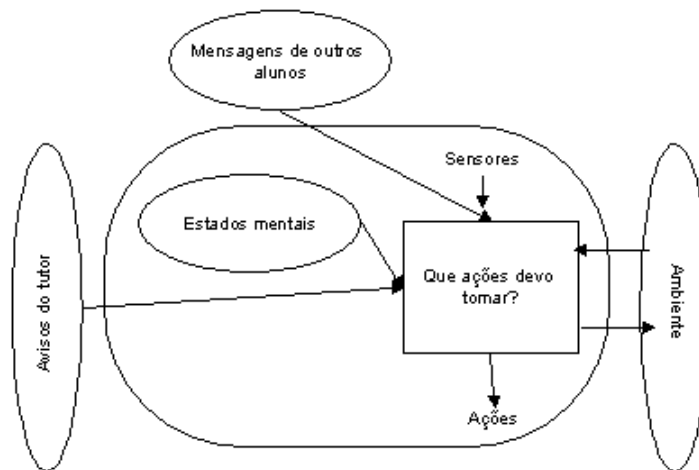


FIGURA 8- Componentes do Modelo do Aluno do MCOE.

O ambiente AME-A [DAM 98], [PER 98] e [PER 98a], também desenvolvido no Grupo de Inteligência Artificial em Educação do Programa de Pós-Graduação em Ciência da Computação da UFRGS, é um ambiente de ensino, para o qual se propõe o estudo e o desenvolvimento de um sistema educacional interativo para o ensino à distância, e o ensino genérico e adaptável às características psico-pedagógicas do aprendiz. Entre as características do ambiente pode-se citar:

- *Aprendizagem Estática*, a qual é utilizada para classificar o aprendiz entre ‘N’ possíveis modelos psico-pedagógicos e para determinar entre as ‘M’ possíveis estratégias de ensino a que melhor se adapta ao mesmo;
- *Aprendizagem Dinâmica*, que é realizada durante a interação do aprendiz e será utilizada para corrigir falsas concepções a respeito do aprendiz inferidas pelos demais agentes atuantes no processo de ensino.

Os principais agentes que fazem parte do processo de ensino do ambiente AME-A, são os seguintes:

O agente *Modela_Aprendiz* identifica o nível de conhecimento do aprendiz, seus objetivos de aprendizagem, motivação e as suas características psico-pedagógicas. Para determinar as características psico-pedagógicas que interferem no processo de ensino, utiliza-se uma taxinomia que classifica os indivíduos em quatro pares opostos, onde é analisado um grupo de características pessoais A partir das definições, determina-se dezesseis tipos diferentes de perfis psicológicos pelas combinações dos quatro pares de valores. O agente *Modela_Aprendiz* analisa e define em qual dos 16 tipos o aprendiz se enquadra, através de um questionário aplicado no início do curso. As informações sobre o perfil do aluno, nível de conhecimento, motivação e nível de profundidade para o objetivo em curso são enviadas ao agente *Seleciona_Estratégia*.

O agente *Seleciona_Estratégia* tem como objetivo selecionar estratégias de ensino para cada modelo de aprendiz, verificar os objetivos de aprendizagem, e informar ao agente *Orienta_Aprendizagem* as estratégias selecionadas. As estratégias podem ser vistas como esquemas de planos que definem formas de apresentar o material instrucional ao aprendiz, ou seja, como gerar uma seqüência de táticas de ensino para apresentar com sucesso um determinado tópico a um determinado aprendiz. As táticas são ações definidas para exteriorizar uma estratégia. Entre as táticas tem-se:

1. Mostrar um exemplo usando uma situação similar;

2. Mostrar uma mensagem com a melhor opção;
3. Mostrar exemplos relacionados sem nenhuma explicação;
4. Mostrar temas que são importantes prestar mais atenção;
5. Mostrar o conteúdo de cada tópico;
6. Quando o estudante não está aprendendo corretamente, mostrar mensagens e avisos para orientá-lo;
7. Mostrar uma mensagem explicando a consequência de suas ações;
8. Mostrar ao aprendiz sucessivas questões para ele poder analisar hipóteses, descobrir contradições e fazer inferências corretas;
9. Com base nos erros do aprendiz, apresentar exemplos análogos para compreensão da solução e determinação dos erros através de outros casos;
10. Simular situações para levar o aprendiz a construir total ou parcialmente os conceitos que estão sendo ensinados.

Uma rede neural mapeia características do aprendiz e características de todas as estratégias de ensino. O agente *Seleciona_Estratégia* muda a estratégia para determinado aprendiz quando, o agente *Modela_Aprendiz* ou o agente *Analisa_Aprendizagem*, informarem alguma alteração na aprendizagem, motivação e personalidade do aprendiz. Caso um determinado aprendiz não esteja aprendendo corretamente, o agente vai tentar uma outra estratégia que não foi adotada inicialmente para este tipo de aprendiz, e assim vai tentando até obter êxito.

O material de ensino (domínio do conhecimento) está armazenado em forma de páginas HTML conforme tipos pré-definidos, por exemplo, vídeos, conceitos gerais, conceitos mais abrangentes, exemplos, mensagens, alertas, gráficos, tabelas, etc. Após a seleção da estratégia, o agente *Orienta_Aprendizagem* irá buscar em um banco de dados o endereço do material a ser apresentado.

O Projeto TRILOGY [NOR 2000], composto por três universidades com interesses comuns de pesquisa em engenharia de tráfego em redes de telecomunicações, propõe um sistema para o treinamento de estudantes pesquisadores através do desenvolvimento de um laboratório virtual. Esse treinamento envolve as atividades básicas de pesquisa envolvendo Colaboração, Informação e Experimentação. Assim, o sistema provê os meios para permitir orientação ou troca de idéias com colegas pesquisadores mais experientes, busca e acesso a documentos, artigos e relatórios de pesquisas e ferramentas ou instrumentos para pesquisa experimental, tais como, software simuladores que possibilitem teste e avaliação de hipóteses experimentais.

O Laboratório Virtual proposto no projeto TRILOGY fornece estes recursos de forma adequada ao nível do estudante pesquisador, ou seja, aos alunos novíços devem ser apresentados artigos introdutórios ou tutoriais, enquanto que aos alunos mais graduados, ou aos pesquisadores experientes devem ser apresentados os avanços mais recentes. Além disso, o sistema sugere informações relevantes ao usuário de acordo com a área de interesse como, por exemplo, identificação de tutoriais relativos à matéria, ferramentas experimentais disponíveis e interação com outros pesquisadores em tópicos semelhantes. Além disso, o sistema provê meios para administrar o uso de recursos escassos, controlando uma agenda

Para tanto, foi empregada uma arquitetura multiagentes constituída por três tipos de Agentes o *Personal Assistant Agent* (PAA), que é instanciado para cada usuário e

tem a função de representar os interesses deste usuário, mediante o sistema, o *Mediator*, que interage com os demais agentes fornecendo informações em um nível de abstração intermediário, como quais os recursos disponíveis no sistema e quais os usuários com interesses comuns, e o *Resource Agent* (RA) que representa e controla os recursos disponíveis no sistema, como mecanismos de busca, controle de instrumentos de pesquisa experimental. O RA permite que o sistema seja largamente expansível, pois qualquer novo recurso pode ser adicionado ao laboratório, com o desenvolvimento do correspondente RA, que o administre e execute as interações do novo recurso com os demais agentes do sistema. Os agentes comunicam-se através de uma linguagem formal e as mensagens entre eles são feitas através de tecnologia CORBA

Dos trabalhos anteriormente descritos, os seguintes aspectos foram considerados na elaboração do JADE:

- Utilização de agentes: com exceção do CALAT e do ELM-ART II, que adota uma arquitetura cliente-servidor, todos os trabalhos anteriormente descritos utilizam a abordagem de agentes. Foi inspirado neles que a arquitetura do JADE adotou esta abordagem. Entretanto a maioria desses trabalhos ou implementa um único agente tutor, como é o caso do ADELE e do MCOE, ou implementa múltiplos agentes como diferentes *threads* de um mesmo programa, como é o caso do AMEA e do SEI. Esta arquitetura faz com que os agentes sejam menos independentes entre si e não permite tirar o máximo proveito que uma modelagem do problema baseado em cooperação e sinergia entre diferentes processos independentes, representados pelos agentes poderia resultar, como é o caso da arquitetura utilizada no TRILOGY.
- Papel dos Agentes: os sistemas descritos anteriormente, que adotam arquitetura multiagentes, implementam agentes que desempenham papel de acordo com uma metáfora das funções de um professor, como o APPEAL e o GIA, ou procuram seguir a arquitetura tradicional de sistemas Tutores Inteligentes, como o SEI e o AMEA, com os agentes desempenhando funções equivalentes aos módulos utilizados nestes sistemas (modelo de aluno, modelo de domínio e modelo de estratégias de ensino) Ou ainda o papel de procuradores, como é o caso do TRILOGY. As três abordagens parecem formas adequadas de dividir o problema do ensino em subproblemas, mas um modelo híbrido que misture as três abordagens ou uma arquitetura que não utilize necessariamente nenhuma destas metáforas poderia eventualmente conseguir melhores resultados.
- Arquitetura interna dos agentes: com exceção do TRILOGY, não foi identificado nos sistemas descritos uma preocupação explícita com a modelagem da arquitetura de cada agente individualmente de modo a fazer com que diversos agentes com diferentes papéis tivessem, na medida do possível, uma mesma arquitetura, possibilitando maior reaproveitamento de código, facilidade de evolução do sistema e independência do modelo de comunicação e de cooperação da arquitetura interna dos agentes, tornando o sistema mais robusto.
- Comunicação entre os agentes: os sistemas TRILOGY, GIA, AMEA e SEI fazem referência explícita ao uso de um modelo formal de linguagem de comunicação entre os agentes, a linguagem KQML, sendo que no SEI a implementação da troca de mensagens é feita através de mecanismos de RMI

da linguagem JAVA e no TRILOGY é utilizado CORBA. No atual estado da arte, essas parecem ser as duas formas mais indicadas, pelo fato de permitirem a implementação das mensagens em forma de objetos de classes e transmitir estes objetos entre os agentes, sendo que a primeira é mais acessível mas a segunda é mais robusta e está se tornando um padrão de fato. Não foi identificado, a definição de um conjunto de mensagens ou de uma ontologia para agentes pedagógicos que pudesse ser empregada diretamente em uma nova arquitetura.

- Retenção dos dados dos modelos de conhecimento dos agentes: com exceção do sistema AMEA, nenhum outro faz referência explícita à utilização de banco de dados como ferramenta para a retenção dos dados sobre o modelo do aluno, modelo do domínio e o modelo das estratégias de ensino. Um modelo relacional adequado de dados, mapeado a partir do modelo de conhecimento dos agentes, abrangendo o Modelo de Domínio, o Modelo de Aluno e o Modelo de Estratégias, implementado em um banco de dados comercial, libera os agentes da tarefa de retenção dos dados, agrega robustez ao sistema e facilita a implementação de ferramentas de autoria e de administração dos cursos.
- Independência de domínio: os sistemas CALAT, TRILOGY e o AMEA parecem ser os que melhor apresentam uma arquitetura independente do domínio de conhecimento referente a matéria a ser ensinada. Nesses sistemas o material instrucional encontra-se totalmente fora da arquitetura interna do sistema, sendo que, no primeiro, ele é referenciado na forma de *links* em páginas HTML, o segundo, utiliza ferramentas de busca na rede ou recursos independentes controlados por seu próprio agente, e o terceiro, em referências contidas em banco de dados. Esta alternativa parece mais adequada aos objetivos do JADE pelo fato de facilitar o processo de implementação de novos cursos, através da manipulação do banco de dados.
- Modelo de aluno: de um modo geral, os sistemas utilizam um modelo de aluno baseado em *overlay*, isto é através do modelo de domínio é assinalado, à medida que a sessão transcorre, quais conteúdos ou quais objetivos pedagógicos o aluno alcançou. Este acompanhamento é feito através da performance do aluno nos exercícios feitos. Os sistemas SEI e AMEA utilizam modelo de estereótipos para classificar os alunos. Para tanto, fazem uso de um questionário inicial a fim de classificar os alunos em um dos estereótipos previamente modelados. Nenhuma das duas abordagens parece ser suficientemente adequada para implementar uma metodologia de ensino que seja adaptativa e dinâmica, e o uso de questionário, além de encontrar muita resistência e desmotivação, pode levar a um diagnóstico impreciso. O sistema APPEAL utiliza uma forma pela qual o modelo de aluno além de registrar os tópicos e os exercícios executados, classifica o aluno de acordo com a sua performance e o seu comportamento em cada uma das atividades. Este modelo seria mais adequado, principalmente se os mecanismos de observação do comportamento e de diagnóstico forem aprimorados ao ponto de monitorar, não só a performance nos exercícios, mas todas as ações do aluno, tais como o tempo despendido em cada tópico, o nível de ajuda solicitado, o tipo de ferramenta do sistema que o aluno utiliza, os erros mais freqüentes e outros.

- Modelo de estratégias: o sistema CALAT divide as atividades pedagógicas em três tipos: explanação, exercícios e simulação, sendo que cada unidade a ser ensinada é composta por uma seqüência destas três atividades. Este modelo vem sendo também utilizado pelo Eletrotutor. O ELM-ART II utiliza também a idéia de transmissão de conceitos teóricos, a apresentação de exemplos e a proposição de problemas. O ADELE possui um modelo de ensino baseado em casos e em resolução de problema. O material didático é constituído por um arquivo separado do sistema que pode ser baixado pela rede ou recebido em CD. O MCOE é concebido para o desenvolvimento de habilidades em um conteúdo específico e utiliza simulação para desenvolver habilidades e atitudes. O APPEAL utiliza um modelo de intervenção multicamadas, no qual o nível de intervenção do sistema varia desde um simples encorajamento até um nível de demonstração, de acordo com os dados do modelo do aluno. No sistema AMEA as estratégias são tratadas como esquemas de planos previamente estabelecidos que definem formas de mostrar o material. Uma rede neural mapeia as características do aprendiz e as características das estratégias de ensino a fim de estabelecer qual a melhor estratégia para cada aluno específico. O TRILOGY é um laboratório virtual, por tanto sua metodologia de trabalho consiste em prover recursos e sugerir atividades ao aluno pesquisador. Nenhum dos sistemas descritos possui recursos que permitam ao autor do curso definir as estratégias durante a fase de projeto e implementação do curso, propriamente dito. Isto é, nenhuma das arquiteturas é independente do modelo de estratégias, nem permite, pelo que parece, a utilização de diferentes conjuntos de estratégias em diferentes cursos.

3 O JADE

A partir dos objetivos anteriormente descritos, com base nas premissas e nas hipóteses levantadas, e levando em conta os trabalhos correlatos descritos anteriormente, foi desenvolvido um conjunto de recursos para facilitar o desenvolvimento e a implementação de ambientes computacionais para utilização como instrumento de Educação a Distância, com características que atendam a estas premissas. Esse conjunto de recursos ou, essa infra-estrutura (*framework*) como costuma ser chamada, foi denominada JADE que é um mnemônico para *Java Agent framework for Distance learning Environments* ou *infra-estrutura de agentes Java para ambientes de ensino a distância*.

O propósito do JADE é, ao mesmo tempo, prover uma infra-estrutura básica de código em linguagem de programação Java, que vem sendo utilizada em larga escala para a implementação de agentes distribuídos, em diversas aplicações, assim como proporcionar uma metodologia para o desenvolvimento de aplicações de Educação a Distância, contemplando uma metodologia adequada de ensino, e que possa ser aplicada para desenvolver aplicações para um razoável espectro de tipos de conteúdo.

Dentro desta perspectiva, os pontos considerados na definição da infra-estrutura proposta foram:

- utilização de um método formal de modelagem empregado para desenvolver o projeto do JADE;
- a definição da metodologia de ensino a ser empregada pelo JADE, de acordo com princípios psicopedagógicos adequados;
- a definição da arquitetura global e da arquitetura interna dos agentes que o constituem de forma a viabilizar estes princípios;
- a definição de mecanismos de cooperação entre estes agentes;
- os mecanismos de comunicação que suportam a cooperação entre os agentes;
- o projeto físico e a implementação do sistema;
- a definição de uma metodologia para implementar ambientes de ensino com o JADE.

Dessa forma, os princípios que nortearam cada um desses pontos e que foram utilizadas como premissas básicas neste projeto, de acordo com os objetivos estabelecidos, serão discutidas a seguir.

3.1 Metodologia de análise e modelagem

O projeto e o desenvolvimento de ambientes de ensino, como todo o sistema complexo, requer a utilização de metodologia e instrumentos adequados. Segundo [BRE 98], grande parte dos sistemas orientados a arquiteturas multiagentes existentes correntemente utilizam metodologias e ambientes de desenvolvimentos orientados a objetos. Os métodos clássicos de engenharia de software orientados a objetos, no entanto não podem ser transferidos para a abordagem de agentes sem algumas adaptações.

Quatro diferentes etapas caracterizam o modelo tradicional de desenvolvimento de sistemas: a análise, o projeto, a implementação e o teste.

Uma especificação geral do problema a ser modelado é feita na fase de *análise*. O objetivo desta etapa é a criação de um modelo que defina com precisão as tarefas a serem desempenhadas pelo novo sistema e o que ele deverá fazer para atingir os seus objetivos. A fase de análise não considera em detalhes como a tarefa é especificamente resolvida.

No desenvolvimento de ambientes de ensino, esta fase pode ser caracterizada pela análise do domínio ao qual se destina o sistema, a definição do escopo do sistema, dos objetivos educacionais, da metodologia e das técnicas mais adequadas ao domínio e a população-alvo a que se destina o sistema e as formas de avaliação.

A fase seguinte é a fase de *projeto*. Nesta etapa, o modelo de análise é utilizado para completar o real projeto do sistema, no qual devem ser definidas a arquitetura do sistema, as unidades funcionais, as estruturas de dados e os algoritmos empregados. No domínio da Educação, esta fase caracteriza-se pelo desafio do projetista do sistema em transformar em especificação formal todas as necessidades levantadas na fase anterior, na qual predomina a preocupação da eficiência e eficácia do sistema do ponto de vista educacional.

Na fase de *implementação*, as linguagens de programação e outras ferramentas, mais adequadas à especificação do projeto devem ser escolhidas e utilizadas para converter os conceitos desenvolvidos em realizações práticas.

Finalmente uma etapa completa de teste e validação do sistema deve ser realizada. Em sistemas de ambientes de ensino a validação do sistema deve ser feita com os instrumentos e a metodologia adequados para avaliar se os objetivos educacionais estabelecidos podem ser alcançados através da utilização do sistema pela população-alvo para a qual ele foi designado.

A questão que nos ocorre é qual a melhor abordagem ou a metodologia mais adequada para levar a termo um projeto de desenvolvimento de ambientes inteligentes distribuídos de ensino. Um dos principais objetivos do JADE é, justamente, facilitar o desenvolvimento de cada uma dessas etapas, fornecendo uma infra-estrutura básica na qual o projetista é poupado de algumas destas tarefas, que seriam comuns a um largo espectro de tipos de ambientes de ensino. Para tanto, cada uma destas etapas foi realizada na concepção do JADE, através de uma metodologia formal de análise e projeto.

3.1.1 Análise orientada a agentes

Com a premissa de utilização de arquiteturas de sistemas multiagentes na construção destes ambientes, a análise orientada a objetos [COA 93] apresenta-se, à primeira vista, como uma boa alternativa. A natureza dos sistemas de agentes de software podem ser modelados como sistemas de objetos. Seus objetos, os agentes, consistem de atributos e métodos e comunicam-se entre si invocando métodos ou enviando mensagens, e fazem uso de conceitos clássicos de orientação a objetos, tais como herança, encapsulamento de dados, agregação, etc. Os métodos mais tradicionais, tais como a análise e o projeto estruturados, baseiam-se em programas de lógica funcional, que não representam adequadamente muitos aspectos de um sistema multiagentes.

A técnica de modelagem orientada a objetos utiliza modelos gráficos com o objetivo de simplificar a tarefa de concepção de estruturação do sistema para o projetista. Estes gráficos constituem o *Modelo de Objetos*, que representa a estrutura

estática de todos os objetos, suas propriedades e seus relacionamentos, o *Modelo Dinâmico*, que representa a dinâmica dos processos, e um *Modelo Funcional* que descreve o fluxo de dados dentro do sistema e a sua estrutura.

Entretanto, embora agentes sejam em muitos casos representados como objetos, eles possuem determinadas características que os distinguem dos sistemas orientados a objetos clássicos. Por isso, os métodos puramente orientados a objetos têm apenas uma limitada adequação ao uso no desenvolvimento de sistemas baseados em agentes. Segundo [BRE 98] os trabalhos mais importantes de modelagem de sistemas baseados em agentes concluem que agentes têm uma estrutura interna muito mais complexa do que a estrutura tradicional de objetos. Eles não consistem somente em atributos e métodos, mas também em estados mentais e conceitos tais como planos ou objetivos.

A estrutura interna de um agente, desta forma, consiste de um componente funcional e também de um número de padrões de comportamento. Os dois aspectos devem ser levados em consideração na modelagem. Um agente tem mais similaridade com um subsistema de métodos orientados a objetos do que com um simples objeto.

O processo de modelagem deve, portanto, levar em conta o papel do agente dentro da organização. Em contraste com o conceito clássico de objetos que são passivos, agentes são unidades ativas e autônomas. Eles podem agir independentemente e seguir seus próprios planos e objetivos. Enquanto que objetos são ativados através do envio de mensagens ou invocação de métodos, um agente pode, a qualquer momento, decidir se, e de que modo, responde a uma mensagem específica. Uma metodologia de projeto de sistemas orientados a agentes deve prover modelos que representem este padrão de comportamento.

A forma de comunicação entre agentes é outro fator significativo de diferenciação entre arquiteturas orientadas a objetos e arquiteturas orientadas a agentes. Em sistemas orientados a objetos a comunicação é realizada em baixo nível, constituindo-se basicamente na troca de mensagens por meio das funcionalidades providas pelos objetos, dentro do princípio clássico cliente-servidor. Agentes, por sua vez, possuem um nível de comunicação significativamente mais poderoso, que utiliza protocolos complexos de comunicação e estruturas de diálogo com muito mais conteúdos do que a troca de mensagens das estruturas de objetos. Por isso, os modelos dinâmicos dos métodos tradicionais de modelagem orientada a objetos não oferecem recursos suficientes para modelar mecanismos como os protocolos de cooperação e as estratégias construídas sobre os protocolos de comunicação entre os agentes.

A fase de análise da construção de um sistema orientado a agentes consiste de três modelos: o *Modelo Organizacional*, o *Modelo do Agente* e o *Modelo de Cooperação*.

- *Modelo organizacional* representa os relacionamentos estáticos entre agentes ou entre classes de agentes, podendo consistir em princípios clássicos do modelo de objetos, como herança ou agregação, ou ser baseado nos papéis associados a cada agente dentro da organização.
- *Modelo de Agente* contém a estrutura interna do agente, consistindo dos componentes clássicos do sistema baseado em objetos, os atributos e os métodos, bem como dos padrões de comportamento e as intenções do agente.
- *Modelo de Cooperação* consiste na interação e no processo de cooperação entre os agentes. Basicamente deve ser formado pelo conjunto de mensagens e pelo processo de cooperação e de comunicação entre os agentes.

O Modelo de Cooperação equivale ao modelo dinâmico dos métodos clássicos, entretanto com a diferença de que os processos internos dos agentes, denominados *Ciclo de Vida* do agente fazem parte do Modelo de Agentes e não do Modelo de Cooperação. Os passos para a criação do Modelo de Cooperação incluem *Objetivos de Cooperação*, os *Tipos de Mensagens* e os *Protocolos de Cooperação*.

[KIN 95]propõe dois graus de abstração no processo de modelagem de sistemas orientados a agentes: a Visão Externa e a Visão Interna.

O *Modelo Externo* modela os agentes, suas tarefas e responsabilidades, as informações requeridas pelos agentes e suas interações com outros objetos. O *Modelo Interno* representa a arquitetura do agente, incluindo seu comportamento e suas intenções. Esta divisão traz como principal vantagem separar a arquitetura dos agentes da arquitetura do sistema, e conseqüentemente, o modelo externo é completamente independente da forma dos agentes e vice-versa.

Dois *Modelos Externos* são definidos por [KIN 95]: o *Modelo de Agentes* e o *Modelo Interativo*. O *Modelo de Agentes* descreve os relacionamentos estáticos entre os agentes em forma de uma hierarquia de agentes. O *Modelo de Interação* representa graficamente as responsabilidades, os serviços e os processos de comunicação e de cooperação, bem como relacionamentos de controle entre os agentes. A estrutura de informação e os protocolos fazem parte do *Modelo de Interação*.

Os *Modelos Internos* propostos por [KIN95]são o *Modelo de Crenças*, o *Modelo de Objetivos* e o *Modelo de Planos*. O emprego desta abordagem é particularmente adequada para a análise e projeto de sistemas formado por agentes que empregam arquiteturas BDI (*Belief, Desire, Intentions*) [RAO 95].

O *Modelo de Crenças* representa o conjunto de crenças que o agente possui sobre si mesmo e sobre o seu ambiente. O *Modelo de Objetivos* descreve os objetivos que um agente busca e os eventos aos quais ele responde. O *Modelo de Planos* representa os planos que um agente possui para atingir seus objetivos, através de diagramas de transição de estados.

Outra abordagem de análise e projeto de sistemas baseados em agentes é proposto por [WOO99]. Segundo o autor, o estágio de análise é constituído pelas seguintes etapas, que são desenvolvidas recursivamente até que um modelo forma completa, que satisfaça às necessidades do projeto, seja obtido:

- Identificação dos Papéis no sistema: como resultado, é obtido um modelo prototípico de papéis com uma relação dos papéis-chave que ocorrem no sistema, com uma descrição informal de cada um;
- Para cada Papel, identificar e documentar os Protocolos associados a ele, gerando um Modelo de Protocolos (*Protocol Model*). Estes Protocolos são os padrões de interação que ocorrem no sistema, entre os vários Papéis;
- Utilizando o Modelo de Protocolos, elaborar o Modelo de Papéis: como resultado, é obtido um modelo completo e elaborado de Papéis que documenta os papéis-chave que ocorrem no sistema, suas permissões e responsabilidades, bem como os protocolos dos quais fazem parte.

Na fase de projeto, o objetivo básico é, normalmente, transformar os modelos abstratos obtidos durante a fase de análise em modelos com suficientemente detalhados de abstração que permitam a implementação do sistema. Neste caso, porém, o objetivo é

transformar os modelos, obtidos na análise, em um nível de abstração no qual uma técnica tradicional, tal como Orientação a Objetos [FUR 98] possa ser aplicada.

Em outras palavras, o processo de análise projeto orientado a agentes preocupa-se em como a sociedade de agentes coopera para realizar os objetivos do sistema e o que é requerido de cada agente individualmente para tal. De fato, como um agente realiza seu serviço é do escopo da definição do modelo interno do agente. Para isso, pode ser aplicado o método tradicional de análise e projeto orientado a objeto, chegando-se ao nível de abstração que permita a implementação dos agentes.

A etapa de projeto, na metodologia orientada a agentes, proposta por [WOO 99], envolve os seguintes passos:

- Criação de um Modelo de Agentes, (*Agent Model*) que agrega os papéis em tipos ou classes de agentes em uma hierarquia e documenta as possíveis instâncias de cada agente;
- Desenvolvimento de um Modelo de Serviços (*Services Model*), a partir dos protocolos e das propriedades dos Papéis definidos na fase de análise;
- Desenvolvimento de um Modelo de Conhecimento (*Acquaintance Model*) a partir dos Modelos de Interação e do Modelo de Agentes.

Mais adiante, no item 3.3, será discutido como estes princípios foram aplicados na elaboração do JADE, a partir dos princípios pedagógicos definidos, que serão discutidos a seguir.

3.2 Princípios pedagógicos

Faz parte dos objetivos deste trabalho a adoção de um paradigma sobre Educação segundo o qual todos os recursos implementados, a serem utilizados nos cursos resultantes do JADE, contemplem métodos e técnicas pedagógicas aderentes a este paradigma. Os aspectos de ordem teórica serão discutidos aqui, neste item. Posteriormente, quando for apresentada a técnica de autoria de cursos, empregando o JADE, esses aspectos voltarão a ser abordados, novamente, de forma prática.

Conforme já foi discutido, a atividade educacional proposta por um ambiente computacional a ser utilizado como ferramenta para Educação a Distância não deve incorrer em alguns erros cometidos no ensino tradicional. Além disso, deve levar em conta suas peculiaridades, especialmente, no que se refere à natureza dos seus objetivos e às características da sua clientela, predominantemente, constituída por jovens ou adultos com formação diversificada.

Na bibliografia consultada, foram encontrados alguns preceitos teóricos que serão resumidos a seguir.

Paulo FREIRE [FRE 77], em sua obra **Pedagogia do Oprimido**, coloca o conceito de "educação bancária", quando diz que: .

“A narração, de que o educador é o sujeito, conduz os educandos à memorização mecânica do conteúdo narrado (...) Desta maneira, a educação se torna um ato de depósitos, em que os educandos são depositários e o educador o depositante. (...) Na visão bancária da educação, o saber é uma doação dos que se julgam sábios aos que julgam nada saber”.

Esta concepção de educação é transmitida, geração após geração, de professor para aluno. Cada professor tem a tendência de adotar, de forma intuitiva, como modelo, os mestres que teve durante a sua vida de estudante, perpetuando assim um modelo de educação arcaico, baseado numa concepção de homem como um ser que é considerado inserido num mundo que irá conhecer através de informações que lhe serão fornecidas e que se decidiu que seriam as mais importantes e úteis para ele. É um receptor passivo até que, repleto das informações necessárias, pode repeti-las a outros que ainda não as possuem, assim como pode ser eficiente na sua profissão. [MIZ 86].

[RAM 85]identifica as características deste modelo de educação, dizendo que o professor, representante dos pais e da sociedade, seria o homem que havia captado nos livros o patrimônio cultural da humanidade ali armazenado; e havia se adestrado na transferência deste saber à inteligência do aluno. Além disto, seria sua missão ensinar aos alunos a aceitação das leis e normas que sustentam a vida social e a submissão à autoridade, defensora e guardiã dessas normas.

A educação tradicional gira em torno do professor. É ele quem fala, explica, atua e decide. Ele define o que se deve aprender, estabelece o ritmo do aprendizado, o horário de trabalho e as normas disciplinares. Sua autoridade é autocrática.

O aluno é um mero receptor, limita-se a ouvir (ou fingir que ouve), para gravar na memória verdades dogmáticas, descobertas por outros mais inteligentes do que ele, e tornar-se, desse modo, capaz de repeti-la nos exames. Deve ser dócil, obediente e passivo, deixando-se guiar por aqueles que detêm a autoridade, representada na escola pelo professor.

[RAM 85]resume dez pontos que caracterizam a educação dita tradicional:

- É diretiva, isto é, o professor indica a matéria que deve ser ensinada e como deve sê-lo.
- Os exames são o melhor sistema de seleção escolar.
- Os exames são o princípio, meio e fim da educação.
- Aprende-se só aquilo que é ensinado.
- Processo de aprendizagem é cumulativo.
- Saber é dogmático.
- Rigor científico tem mais importância que as suas conclusões.
- O aluno mais promissor é aquele que repete, com exatidão, os conceitos aprendidos.
- O nível de um estabelecimento é medido pelo rigor de sua seleção.
- O aluno não é considerado uma pessoa mas sim um objeto manipulável por prêmios e castigos.

No pensamento moderno, o homem é considerado dentro do seu contexto social e o seu meio ambiente. [RAM 85]caracteriza o homem moderno como organismo inteligente que atua num meio social. O homem é, portanto, atividade dentro de certas circunstâncias que o condicionam. O homem não é, simplesmente, um animal que possui inteligência, mas um ser continuamente perfectível. Mesmo que seja física e cronologicamente adulto, sempre pode tornar-se mais homem, mais responsável e mais equilibrado.

Carl ROGERS [ROG 77], um dos mais destacados representantes da Psicologia Humanista ou Fenomenológica, coloca as suas proposições acerca do homem e das suas possibilidades, entre as quais destacam-se:

- Todo indivíduo existe num mundo de experiência do qual é o centro e que está em permanente mudança.
- Organismo reage ao campo perceptivo, tal como este é experimentado e aprendido. Este campo é para o indivíduo a sua realidade.
- Organismo reage ao campo fenomenal como um todo organizado.
- Melhor ângulo para a compreensão da conduta é a partir de um quadro de referência interno.

Para ele, a realidade é um fenómeno subjetivo, pois o ser humano reconstrói, em si, o mundo exterior, partindo de sua percepção, recebendo os estímulos e as experiências, atribuindo-lhes um significado.

Dentro desta concepção do homem e seu mundo, [RAM 85]propõe as características de um modelo de educação que parece mais adequada às propostas de um modelo de Educação a Distância, dentro daqueles princípios anteriormente apontados:

1. **Utiliza métodos científicos.** Os novos métodos baseiam-se nos conhecimentos da genética, da biologia, da psicologia e da sociologia. Além disto, foram concebidos cientificamente, isto é, a partir de métodos experimentais.
2. **Métodos vitalistas.** " Educar é ensinar a viver" para a nova educação. O essencial é conseguir o desenvolvimento das potencialidades do aluno e não a simples adaptação ao meio, como queria a pedagogia antiga.
3. **Educação ativa.** Só se pode qualificar de genuinamente ativos, aqueles métodos que levam o aluno a realizar ações de investigação e de criação por própria iniciativa e não só ato de mera repetição das indicações do professor. PIAGET apud [MIZ 86], em seus estudos, demonstra que o ato de aprender exige uma ação. Diz ele que a inteligência consiste em executar e coordenar ações ainda que em forma interiorizada e reflexiva.
4. **Educação personalizada.** A cada dia se torna mais inaceitável um ensino maciço, dirigido a um hipotético aluno médio, obrigando os mais dotados a uma irreparável perda e, abandonando à própria sorte os menos capazes. Por ser único, cada homem tem seu ritmo próprio de assimilação e de crescimento. Isto exige uma educação própria, personalizada.
5. **Educação democrática.** Democracia, mais que uma forma de governo, é uma forma de viver em sociedade, por isto, constitui-se, em si mesma, uma forma de educação, onde o ato de viver constitui-se um constante exercício de liberdade e de respeito aos seus limites. A educação democrática exige, portanto, que se eduque "na liberdade" e "para a liberdade".
6. **Educação flexível.** Alvin TOFFLER, apud [RAM 85]chama "conflito do futuro" à desorientação desastrosa que se provoca nos indivíduos ao obrigá-los a uma mudança excessiva, dentro de um lapso de tempo muito curto, e essa enfermidade da mudança nasce da indiferença entre a velocidade da transformação do meio e a rapidez limitada da reação humana. Essa nova condição de nossa sociedade exige, evidentemente, uma educação nova, mais dinâmica e mais flexível. Numa sociedade móbil, a educação não deve se limitar a ensinar o que os outros fizeram, mas ensinar a fazer o que ainda não foi feito.
7. **Educação permanente.** Durante muito tempo, acreditou-se que o ser humano atingia o máximo de sua inteligência aos treze anos de idade, e que a idade mental do indivíduo médio já se fixava com esta idade. Só na segunda metade deste século, começou-se a falar em educação permanente, estudando-se a psicologia do adulto e a adaptação de métodos de ensino à realidade destes. Atualmente, considera-se a necessidade da educação permanente pela complexidade da vida na sociedade

contemporânea, pela grande soma de conhecimentos necessários à vida inteira, pela constante mudança do ambiente social e dos conhecimentos, e, principalmente, pela possibilidade de a pessoa aperfeiçoar-se a vida inteira.

Na obra de Jean PIAGET, encontram-se também alguns subsídios importantes para a formação de um paradigma orientador da questão da metodologia empregada no desenvolvimento de recursos humanos. [MIZ 86] sintetiza a abordagem Cognitivista-Constructivista que tem em PIAGET o seu maior teórico. Para ele, o indivíduo é considerado como um sistema aberto, em reestruturações sucessivas, que busca um estágio final, nunca alcançado por completo. O conhecimento é considerado como uma construção contínua, a passagem de um estado para o seguinte é sempre caracterizado por formação de novas estruturas que não existiam anteriormente no indivíduo, o conhecimento humano é essencialmente ativo.

O processo educacional, consoante à teoria de desenvolvimento e conhecimento, tem um papel importante ao provocar situações que sejam desequilibradoras para o aluno, desequilíbrios estes, adequados ao nível de desenvolvimento em que se encontram, de forma que seja possível a construção progressiva das noções e operações, ao mesmo tempo que a criança vive intensamente cada etapa de seu desenvolvimento.

O ensino que seja compatível com a teoria piagetiana tem que ser baseado no ensaio e erro, na pesquisa, na investigação, na resolução de problemas por parte dos alunos, e não em aprendizagem de fórmulas, nomenclaturas e definições.

Esta abordagem difere de forma acentuada da abordagem comportamentalista, tendo implicações outras para o ensino. Para PIAGET, apud [MIZ 86], o conhecimento progride mediante a formação de estruturas e isso nega o mecanicismo de justaposição em que se baseiam os comportamentalistas e os que advogam o ensino tradicional.

[MIZ 86] coloca ainda que não existe um modelo pedagógico piagetiano, o que existe é uma teoria de conhecimento. Entretanto, pode-se destacar alguns pressupostos básicos que servem de base para um planejamento das situações de ensino:

- A ação do indivíduo é o centro do processo, sendo baseado, essencialmente, na investigação.
- Trabalho em equipe adquire consistência teórica.
- Ambiente e o material de ensino precisam ser desafiadores e promover a motivação para a investigação.
- A utilização de recursos audiovisuais não é suficiente para desenvolver atividades operatórias. As experiências não devem ser feitas na frente dos alunos e sim por eles.
- Ensino programado leva a aprender mas não a criar, salvo se a programação for feita pelo próprio aluno.
- Um método de ensino individualizado só é coerente se respeitar o ritmo próprio, modo de agir e de criar do aluno, implicando programas, técnicas e horários suficientemente flexíveis e adaptáveis às condições dos alunos.

Da mesma forma que o Constructivismo Piagetiano, a abordagem do Processamento da Informação trabalha com a concepção de representação mental, tendo como objeto de estudo primordial a gênese, o desenvolvimento e as transformações destas representações [EYS 94] e [SIM 81]. A Psicologia Cognitiva do Processamento da Informação diverge em alguns pontos do Constructivismo, embora compartilhe alguns pressupostos básicos, por exemplo, a idéia de que o homem é um agente sobre os seus

pensamentos, emoções e sentimentos, assim como a ênfase dada ao método científico e à retomada do psiquismo humano como objeto de estudo, negligenciado pela Psicanálise e pelo Comportamentalismo.

A Psicologia Cognitiva surgiu em meados da década de 60', sendo considerada uma Psicologia Pós-Comportamentalista, pelo seu caráter empírico, teórico, metodológico e científico, herdeira deste, mas contrapondo-se ao Comportamentalismo no sentido de não limitar seus estudos aos modelos S - R ou S - O - R. Ela busca, pela observação sistemática do comportamento, pela construção de modelos axiomatizados, pela engenhosidade experimental e até mesmo pela simulação computacional advinda da sua ciência irmã, a Inteligência Artificial, estudar a vida mental e prever como se dão os processos mentais e qual a arquitetura destes. A Psicologia Cognitiva tem assim o propósito de desvendar o que ocorre na "caixa-preta" do psiquismo humano [MAD 89].

Os seres humanos passaram neste momento a ser vistos como organismos recebedores, emissores e tratadores de informação e que, portanto, possuem processamentos que apresentam características específicas, possíveis de serem explicadas e preditas. Nessa nova vertente da Psicologia, identifica-se o ser humano como um sujeito ativo no seu ambiente e que possui mecanismos de auto-regulação (cibernéticos) e inclusive déficit nestes mecanismos e naqueles de tratamento da informação. A tarefa assumida pelo psicólogo cognitivo, passa a ser então de caráter ecológico: descobrir o funcionamento cognitivo dos indivíduos com fins de aprimorar as habilidades destes. Assim sendo, há o interesse em ciência básica, mas que estão sendo transformados em intervenções de caráter prático.

Segundo [EYS 94], a Psicologia Cognitiva preocupa-se com o processamento da informação e inclui uma variedade de processos como a atenção, a percepção, a aprendizagem e a memória. Está também preocupada com as estruturas e as formas de representações envolvidas na cognição. A grande diferença entre a abordagem adotada pelos psicólogos cognitivistas e os comportamentalistas é que os primeiros estão interessados em identificar em detalhes o que acontece entre o mecanismo de estímulo e de resposta de comportamento estudados pelos últimos.

Pode-se dizer, deste modo, que a Psicologia Cognitiva propõe-se a elaborar um estudo científico e em profundidade da mente humana sem, no entanto, reduzir toda a sua complexidade, inerente ao seu próprio objeto de estudo, que é o psiquismo humano.

Embora ambas as abordagens cognitivistas (construtivista e de processamento da informação) converjam no fato de assumirem como átomos constitutivos do psiquismo as representações mentais, elas divergem quanto ao aspecto de que o Construtivismo, assume a existência de estruturações cognitivas primitivas, as quais se modificam dentro de um esquema de estruturas lógicas previamente definidas e que aparecem em uma seqüência progressiva ao longo do desenvolvimento humano, enquanto que a abordagem do Processamento da Informação acredita que não há estruturas lógicas previamente determinadas, de forma genética [WAI95].

Outra divergência refere-se ao nível de análise impetrado nas duas abordagens. As tentativas de explicação na abordagem de Processamento da Informação são microscópicas, buscando os processos e estratégias mentais dos sujeitos para configurar a representação do mundo que os rodeia, enquanto que a abordagem Construtivista é macroscópica em suas análises, verificando como resultam as interações entre as estruturas cognitivas próprias da fase de desenvolvimento do indivíduo e a realidade própria de cada sujeito.

Ambas as abordagens, no entanto, fornecem subsídios importantes para a concepção de ambientes computacionais de ensino, uma vez que eles passam a ser, neste contexto, o principal instrumento de ação do indivíduo, como aluno, com o ambiente que o cerca.

Como pode ser observado, a partir das diversas teorias apresentadas, a questão dos princípios psicopedagógicos é bastante complexa, e torna-se muito difícil tentar prescrever uma solução ideal que possa ser aplicada a todo e qualquer conteúdo, a cada situação em particular e, até mesmo à preferência pessoal, seja do aluno ou do professor que orienta a utilização ou a implementação de uma ferramenta computacional de ensino.

Por isso, ao definir-se os recursos didáticos e o processo metodológico a serem empregados na elaboração do JADE, foi buscado atender ao preceito anteriormente discutido no item 1.2.4, segundo o qual um ambiente de ensino para ser robusto deve prover diferentes metodologias e empregá-las, de forma circunstancial, de acordo com os Conhecimentos, Habilidades e Atitudes do aluno (CHA).

O JADE utiliza uma metodologia de ensino baseada em um conceito adaptado de [PER 98] segundo o qual, as Estratégias Pedagógicas são constituídas por Táticas, as quais utilizam Recursos Didáticos através dos quais ocorre a interação com os alunos.

Cabe aos especialistas, autores dos cursos, desenvolverem Táticas baseadas nos recursos disponíveis (que podem evoluir com a tecnologia) e definir Estratégias Pedagógicas, segundo as quais, várias combinações de Táticas podem ser utilizadas de formas diferentes. Uma mesma Tática pode, evidentemente ser utilizada em diferentes Estratégias.

Aos autores dos cursos cabe também a incumbência de definir um critério de avaliação vinculado a cada Tática e o peso desta avaliação no contexto da Estratégia como um todo.

Para tanto, a atual versão do JADE prevê a utilização de diferentes recursos didáticos que podem ser empregados de formas diferentes de acordo com a Estratégia. O autor do curso elabora o material didático de acordo com as opções de recursos existentes, que deseja oferecer, bem como desenha diferentes Táticas, de acordo com a maneira de empregar estes recursos. Quanto mais recursos o autor obtiver, mais Táticas podem ser desenvolvidas e um maior número de combinações de formas de utilização dos mesmos pode ser definida, tornando o sistema mais dinâmico e adaptativo.

Cabe ao JADE manusear corretamente estes recursos com cada aluno durante a sessão de ensino, adaptando-se dinamicamente ao aluno, como faria um professor em sala de aula, utilizando métodos apropriados a cada situação, em particular, de acordo com o desempenho do mesmo, com a vantagem, neste caso, de que a adequação do método, utilizando a estratégia mais adequada, é feita de forma individualizada, para cada aluno, e não coletivamente como normalmente ocorre em sala de aula.

Este manuseio é feito através do emprego de um Agente Pedagógico diferente para cada tipo de Tática de Ensino. Os agentes manuseiam o conteúdo e empregam as estratégias de acordo com as informações disponíveis a respeito do aluno. Para isso, os agentes possuem uma Base de Conhecimentos que é constituída por um Modelo do Aluno, um Modelo do Domínio e um Modelo de Estratégias. Esta base de conhecimentos é mapeada para um Modelo Relacional e implementada por meio do emprego de um Gerenciador de Banco de Dados Comercial.

O uso do banco de dados garante a retenção das informações da base de conhecimento dos agentes sem onerá-los com a carga do processamento necessário para o gerenciamento físico da base de dados. O Modelo Relacional da base de dados, empregado pelos agentes do JADE, é descrito em detalhes no capítulo 4.6.

Deste modo, a metodologia de ensino empregada é resultado da sinergia dos agentes pedagógicos cooperando entre si através do emprego de diferentes táticas de ensino, sobre as quais são especializados, sob a coordenação de um agente que manuseia as informações a respeito dos alunos.

O JADE provê uma Ferramenta Administrativa que facilita ao autor, após a criação dos recursos, através de ferramentas de autoria (editor HTML, editor de figuras, imagens, filmes, e editor de textos), a inserção das informações no banco de dados, sendo por este meio definidas as estratégias, os critérios de avaliação e as regras de aplicação das táticas em cada Estratégia.

Além disso, esta ferramenta permite emitir relatórios através dos quais o coordenador do curso pode acompanhar o andamento das atividades dos alunos.

Os Agentes Pedagógicos utilizam os recursos pedagógicos criados pelo autor, durante a sessão de ensino, de forma inteligente e adaptativa, de acordo com as metodologias previamente definidas. Os agentes disponíveis hoje no JADE estão preparados para manusear as seguintes Estratégias:

- Lições – São as diversas formas de organizar os conteúdos teóricos a serem apresentados ao aluno. Para um mesmo conteúdo, podem ser desenvolvidas pelo autor várias lições, utilizando diferentes recursos audiovisuais ou diferentes formas de abordar o tema. O número de tipos de lição para um mesmo conteúdo, fica totalmente a critério do autor do curso.
- Exemplos – São demonstrações práticas dos conceitos de cada lição, através dos quais o aluno pode inferir os conhecimentos que compõem os objetivos da unidade;
- Exercícios – São problemas propostos ao aluno para que ele possa raciocinar aplicando os conceitos a serem aprendidos;
- Salas de interação – Através de salas virtuais de interação ou *chats*, como são conhecidos, o aluno pode trocar informações ou tirar dúvidas com colegas ou professores que eventualmente estejam conectados ao sistema;
- Ajuda – Os recursos de ajuda constituem-se em links para endereços na rede onde o aluno pode encontrar material complementar sobre o tema estudado ou revisar conhecimentos considerados como pré-requisitos para a lição. Vários tipos de ajuda forma considerados como, por exemplo, o acesso a salas de reunião (*chats*) que propicie interação com professores e colegas, endereços na rede que contenham conteúdos pertinentes ou de revisão de pré-requisitos, e uma ferramenta de dicas que o aluno pode recorrer para buscar ajuda.

As diferentes combinações entre os diferentes tipos de lições, exercícios, etc., e a maneira como estes recursos podem ser utilizados pelo aluno, formam as Estratégias que podem ser empregadas pelo JADE. A Estratégia inicial a ser utilizada, é definida pelo próprio aluno, como uma opção pessoal, ou de acordo com a orientação recebida por um professor ou orientador. Não é utilizado pelo JADE nenhum tipo de questionário com a finalidade de estabelecer um diagnóstico inicial a respeito do aluno.

Conforme já foi mencionado, o autor do curso deve criar um repertório de estratégias que contemple a mais variada gama possível de perfis de alunos, de acordo com diferentes graus de Conhecimentos, Habilidades e Atitudes, contemplando desde um método mais diretivo, onde as atividades são prescritas ao aluno, indicando passo a passo quais as atividades que ele deve fazer a cada momento, até uma metodologia totalmente centrada no aluno, onde ele possa ser livre para escolher as atividades que achar mais convenientes, a fim de atingir os objetivos a que se propôs.

3.3 O modelo organizacional

Um dos objetivos específicos deste projeto é propor uma arquitetura para uma sociedade de agentes, adequada às premissas adotadas, a partir de uma metodologia formal de modelagem de sistema orientada a agentes. Espera-se com isso prover o JADE com um modelo de sociedade de agentes adequada aos seus propósitos. Este modelo será apresentado a seguir, neste item.

A literatura a respeito de agentes e de sistemas multiagentes é bastante diversificada, criando, algumas vezes uma certa confusão sobre o conceito de agente, dependendo do enfoque abordado ou do tipo de aplicação. Por isso, antes de tudo, é preciso definir precisamente o que se entende por agente.

Neste projeto, assume-se o conceito de [BRA 97], já discutido anteriormente, segundo o qual, um agente é uma entidade de software que funciona de forma contínua e autônoma, em um ambiente em particular, geralmente habitado por outros agentes, e que seja capaz de intervir no seu ambiente, de forma flexível e inteligente, sem requerer intervenção ou orientação humana constante. De um modo ideal, um agente que funcione continuamente por longos períodos de tempo, deve ser capaz de aprender com a experiência e, se ele habita um ambiente com outros agentes, seja capaz de comunicar-se e cooperar com eles, e ainda mover-se de um local para outro.

Um agente deve, portanto, possuir, em maior ou menor grau, os seguintes atributos: reatividade, autonomia, comportamento colaborador, habilidade de comunicação ao nível de conhecimento, capacidade de inferência, continuidade temporal, personalidade, adaptabilidade e mobilidade:

Conforme já foi discutido, existem duas classes de agentes: os denominados agentes reativos e os agentes cognitivos ou deliberativos. Estes últimos possuem explicitamente um modelo simbólico do mundo ao qual se inserem, e possuem certo grau de capacidade de raciocínio lógico e tomada de decisão como base de sua ação. A modelagem do ambiente é realizada previamente e constitui o principal componente da base de conhecimentos do agente. A capacidade de tomada de decisão dos agentes deliberativos, como reação aos eventos que ocorrem no meio ambiente ou na busca de objetivos ou metas a serem atingidos pelo agente, advém da correta representação interna do ambiente sobre o qual o agente interage.

Embora se saiba das dificuldades para modelar o ambiente e os processos decorrentes das interações existentes nos ambientes de ensino, acredita-se que o emprego de agentes deliberativos seja a melhor solução a ser utilizada. Entretanto, a fim de dar certo grau de reatividade ao JADE, o problema do processo de ensino foi dividido em subproblemas e a cada um desses problemas atômicos é designado um agente especializado.

A consecução dos objetivos pedagógicos, previamente definidos para o ambiente, como um todo, é obtida através da cooperação e do comportamento sinérgico

de um conjunto de agentes, cada um dos quais especializado de alguma forma, na tarefa de auxiliar o aluno a chegar ao seu intento.

As premissas adotadas para prover o JADE de uma estrutura que permita a implementação de uma metodologia de ensino variável e adaptativa, nos moldes descritos anteriormente, dividindo as atividades básicas que, em conjunto, formam uma metodologia de ensino, apontam para um modelo de organização de agentes em que cada uma destas técnicas sejam desenvolvidas de modo especializado por diferentes agentes, de forma autônoma e independente.

Isto se justifica pelas seguintes razões: em primeiro lugar, isto permite que cada uma das técnicas seja aplicada independentemente uma da outra, de acordo com as circunstâncias de cada momento e as características de cada aluno; em segundo lugar, sendo implementados em separado, cada agente se torna extremamente especializado na implementação da técnica, para a qual é encarregado, permitindo que o ambiente possa ser refinado, aprimorando cada uma das técnicas ou mesmo introduzindo novas técnicas com a inclusão de novos agentes.

Isto atende a premissa de que um sistema de ensino deve ser projetado para evoluir, pois através de agentes independentes, manuseando de forma diferente os diversos recursos, é possível introduzir, com o tempo, novos recursos tecnológicos que o agente possa manusear ou, até mesmo, introduzir novos agentes na sociedade, caso um novo tipo de recurso possa ser criado.

Assim, a proposta do JADE é a constituição de um conjunto de indivíduos (agentes) especializados em manusear os recursos definidos para determinadas técnicas de ensino que possam ser utilizados para uma gama de diferentes domínios, e que possuam mecanismos de cooperação com os demais agentes para que, em conjunto, possam atuar no desenvolvimento do curso.

Cada agente possui os conhecimentos e as habilidades que lhe permitam manusear os recursos para o qual foram designados e interagir com os alunos, assim como com os demais agentes que fazem parte da comunidade que compõe o ambiente.

Para isso, os agentes possuem capacidade de comunicação, a fim de interagirem mutuamente, e habilidades especializadas para desempenharem as funções que foram projetadas como, por exemplo, propor exercícios aos alunos e avaliar as suas respostas dando-lhes a realimentação adequada, gerar exemplos adequados ao estado cognitivo de cada aluno, a cada momento, apresentar material instrucional da forma e no momento adequado, e assim por diante.

Evidentemente que, para que os agentes pedagógicos especialistas em cada uma das técnicas possam atuar de forma cooperativa, eles necessitam de mecanismos poderosos de comunicação e de coordenação. Estes mecanismos são obtidos pela arquitetura interna de cada agente, e através de ação de agentes auxiliares especializados em tratar os problemas de comunicação inerentes a arquitetura funcional empregada, e em tratar os problemas de controle de sessão, adaptação das macro estratégias a cada aluno em particular, controle do estado cognitivo de cada aluno, etc.

Para desenvolver o JADE foi adotado o formalismo proposto por [WOO 99], descrito anteriormente no item 3.1, com algumas adaptações para utilizar a simbologia adotada pela UML [FUR 98], por ser uma linguagem de especificação de larga utilização. Como foram estabelecidos os seguintes modelos formais:

- Modelo dos Papéis e suas Permissões e Responsabilidades
- Os Protocolos e o Modelo de Interação
- Modelo de Agentes
- Modelo de Serviços
- Modelo de Conhecimento

A figura 9 mostra o Modelo Organizacional da sociedade de agentes do JADE:

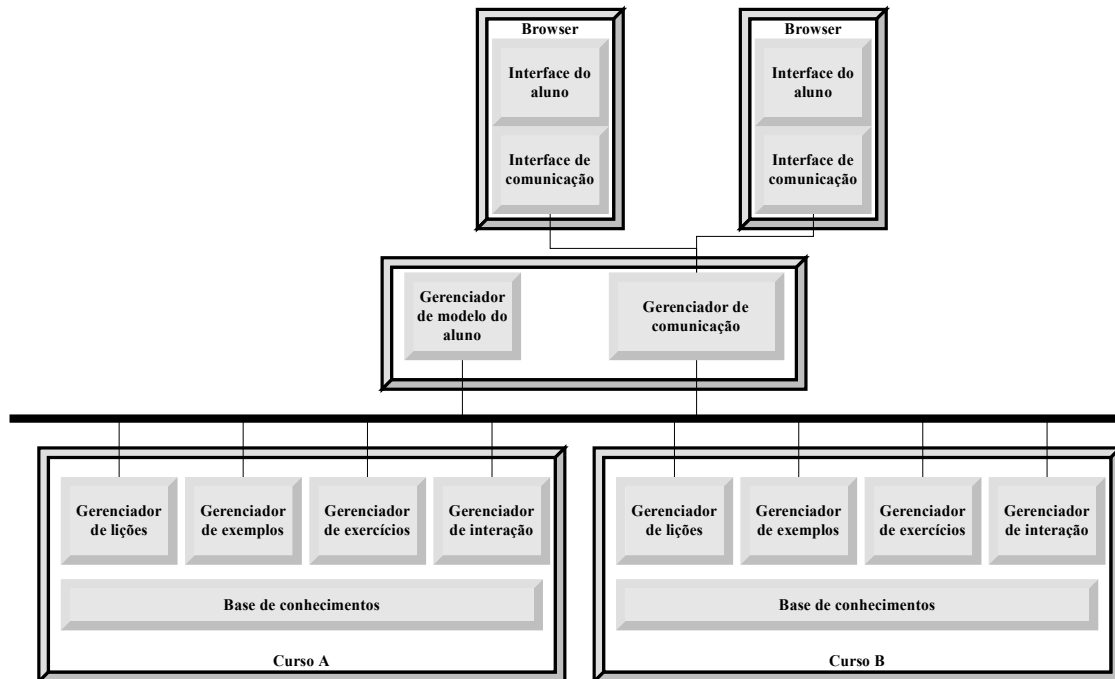


FIGURA 9 - Arquitetura do Modelo Organizacional.

O ambiente contempla uma sociedade de agentes comprometidos com o aprendizado cooperativo, envolvendo os seguintes tipos de agentes:

- *Gerenciador de Conteúdo*: agente responsável pela recuperação do conhecimento do domínio sobre cada ponto a ser apresentado ao aluno;
- *Gerenciador de Exercícios*: agente responsável pela tarefa de propor exercícios ao aluno e avaliar suas respostas;
- *Gerenciador de Exemplos*: agente responsável pela elaboração e controle da apresentação de exemplos;
- *Gerenciador de interação*: responsável por permitir e controlar a interação com outros alunos e professores.

O conjunto de agentes do tipo acima descrito é agrupado para formar um ambiente completo, responsável por um determinado domínio, compartilhando uma mesma base de conhecimentos. Vários grupos de agentes semelhantes a este podem existir em diferentes domínios da rede. Na figura 9 é contemplada uma situação na qual coexistem o ambiente A, com seus quatro agentes pedagógicos básicos, e o ambiente B, especializados em diferentes conteúdos, mas possuindo uma mesma estrutura organizacional de agentes especializados nestas quatro tarefas, porém aplicadas a domínios diferentes.

Cada um destes quatro agentes possui habilidades específicas que lhes permitem desenvolver, com cada aluno, as atividades que lhe são designadas.

Além dos agentes pedagógicos especializados em cada uma das técnicas, outros tipos de agentes são necessários para desenvolver o processo de comunicação e de coordenação:

- *Gerenciador do Modelo do Aluno*: é o agente responsável por construir e manter uma base de conhecimentos que modele o estado cognitivo dos alunos que estejam ou que tenham estado conectados ao sistema. Além disso, é responsável pela inscrição e controle do currículo atendido pelos alunos;
- *Gerenciador de Comunicação*: é o agente responsável pela administração da sociedade de agentes e pelo controle da comunicação entre eles;

Embora não tenham a estrutura interna de agentes, propriamente dito, dois componentes formam a estrutura que é carregada no ambiente do aluno, através da rede quando ele se conecta ao sistema:

- *Interface de Comunicação*: responsável pela comunicação ponto a ponto entre o ambiente do aluno e o ambiente da rede;
- *Interface do Aluno*: responsável pelo controle dos componentes de interação homem-máquina no ambiente do aluno.

3.4 O modelo de cooperação

De acordo com os objetivos específicos deste trabalho, faz-se necessário estabelecer os mecanismos de cooperação e de comunicação, através dos quais a sociedade de agentes interage a fim de atingir seu intento. Estes aspectos serão discutidos a seguir.

O ensino é um processo demasiadamente complexo de ser modelado e manuseado. Por isso, um único agente, para operar, necessitaria de uma grande base de conhecimento e poderia ter problemas de performance. Assim, considera-se que o processo de ensino é tipicamente um problema cuja solução deve ser tratada de forma distribuída, através de agentes especializados, e o problema do processo de ensino e aprendizagem deve ser dividido em subproblemas, cuja solução leve a solução do problema global.

A versão atual do JADE adota o mecanismo de reconhecimento do processo executado pelos demais agentes, por meio do recebimento de mensagens que caracterizam determinados eventos e do compartilhamento de informações oriundas do banco de dados, no qual são armazenadas as informações referentes ao Modelo do Aluno, ao Modelo de Domínio e ao Modelo de Estratégias

Logo, serão discutidos alguns aspectos referentes ao problema da comunicação entre os agentes, onde ocorre o processo de cooperação.

3.4.1 Comunicação

Um dos problemas que atinge o desenvolvimento de sistemas baseados em sociedades de agentes, independentemente da abordagem utilizada na questão da cooperação, é a comunicação entre os agentes. Uma estrutura formal de linguagem de comunicação entre os elementos de uma sociedade é fundamental para permitir a interação social entre estes elementos. Com a finalidade de definir uma forma de cooperação entre os agentes, foi desenvolvido um protocolo de troca de mensagens cuja descrição será discutida mais adiante. Conforme foi discutido anteriormente, a comunicação ocorre através da linguagem KQML

Foi definida a dinâmica de funcionamento do ambiente JADE através de uma seqüência de troca de mensagens entre os diversos agentes que compõem o sistema. As trocas de mensagens são descritas parcialmente nos diagramas mostrados nas figuras 10, 11 e 12.

A figura 10 mostra a dinâmica de funcionamento relativa ao início da sessão e o andamento das unidades correspondentes à primeira lição que é apresentada ao aluno.

Ao conectar-se ao sistema, utilizando um navegador, o aluno deve aceder ao endereço WWW onde encontra-se a página inicial do ambiente, inserindo a URL correspondente àquele endereço. Com isso, o aluno passa a visualizar a página HTML inicial do sistema que contém as informações que o aluno irá necessitar para dar andamento à sessão.

Ao mesmo tempo que o aluno recebe estas informações, um *Java applet* é enviado ao seu computador, que consiste em um programa que é executado pela Máquina Virtual Java existente em seu navegador. Este programa contém os recursos relativos aos denominados *Presentation manager* e a *Communication interface*, descritas anteriormente.

O *Presentation manager* exhibe para o aluno um formulário onde devem ser inseridas algumas informações necessárias para a sua identificação. Ao preencher o formulário, a *Communication Interface* envia estes dados a *Communication interface* remota do sistema, que por sua vez os encaminha ao *Student model manager*.

O *Student model manager*, por seu lado, de posse desses dados, irá buscar em sua base de conhecimentos se este aluno já consta de seus registros. Em caso positivo, irá buscar as informações relativas ao aluno para tomar suas decisões, caso contrário dará início a uma nova instância de aluno inserindo nela os dados de que dispõe.

Neste momento o *Student model manager* envia uma mensagem ao agente *Domain manager* capacitado a atender a demanda do aluno, conforme o julgamento feito pelo *Student model manager* com os dados de que dispõe. O *Domain manager*, por sua vez inicia a sessão de ensino propriamente dita, enviando o material instrucional ao aluno através dos agentes encarregados da comunicação entre o aluno e o sistema, assumindo momentaneamente o controle da interação com aquele aluno específico. Cabe observar que este processo ocorre simultaneamente com diversos alunos.

Diagrama de fluxo de mensagens - fase das lições

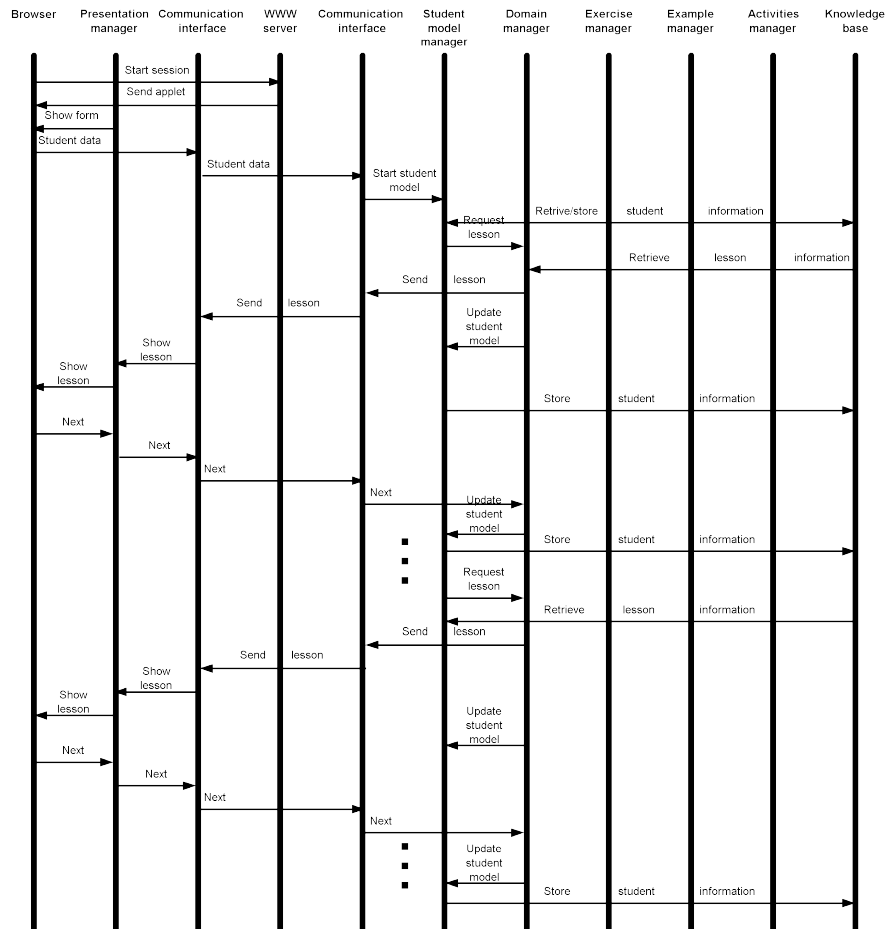


FIGURA 10 – Fluxo de Mensagens na Fase das Lições.

A cada unidade recebida pela interface de comunicação do programa remoto que está sendo executado no ambiente do aluno, o *Presentation manager* monta as telas correspondentes com os textos e os dispositivos necessários para a interação com o agente *Domain manager*. Cabe lembrar mais uma vez que toda a comunicação entre os agentes do sistema e os agentes remotos que rodam no ambiente do aluno ocorre por meio de um par de agentes que fazem o roteamento das mensagens, denominados *Communication Interface*. Esta fase da sessão transcorre com o aluno mostrando ao sistema que pode dar prosseguimento. Este por sua vez envia novo material instrucional e informa ao *Student model manager* até que a unidade seja concluída e a sessão passe para a fase seguinte.

A fase seguinte, no caso do Eletrotutor, será normalmente a fase de apresentação de exemplos ao aluno. Esta fase transcorre através de um fluxo de mensagens similar àquela mostrada na figura 11.

Sempre que o *Student model manager* perceber que o momento é adequado, normalmente após uma seqüência de lições de uma unidade de estudo, ele invoca o *Example manager* para atender àquele aluno, enviando-lhe uma mensagem de requisição de exemplo.

Diagrama de fluxo de mensagens - fase dos exemplos

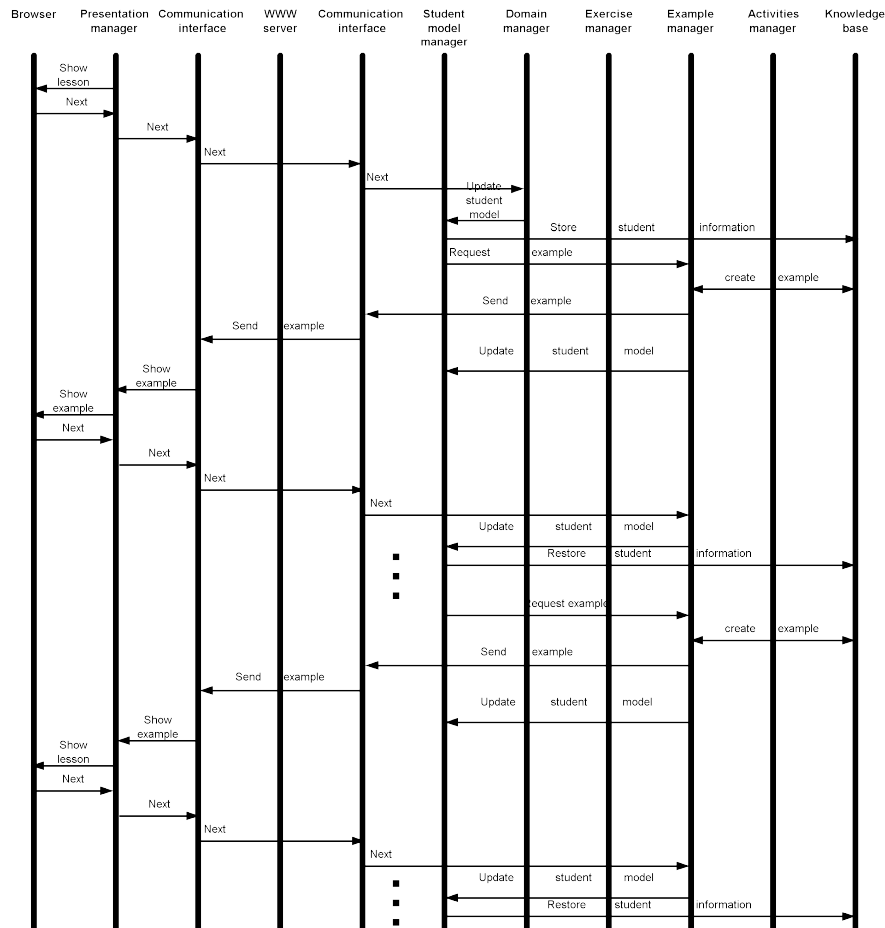


FIGURA 11 – Fluxo de Mensagens na Fase dos Exemplos.

O *Example manager*, ao receber a mensagem cria um exemplo adequado àquele aluno naquela situação particular. Este exemplo é encaminhado ao aluno através do *Communication interface*, da mesma forma que as lições. O *Presentation manager* que está executando no navegador do aluno recebe a mensagem com o exemplo gerado e apresenta-o ao aluno. O aluno, recebendo a mensagem deverá interagir com o sistema solicitando novos exemplos ou pedindo para prosseguir a sessão. De acordo com seu ato, uma mensagem é gerada ao *Example manager* que irá reagir enviando um novo

exemplo ou comunicando ao *Student model manager* que sua tarefa terminou e que ele deverá atualizar seus conhecimentos sobre o aluno e dar prosseguimento a sessão.

Diagrama de fluxo de mensagens - fase dos exercícios

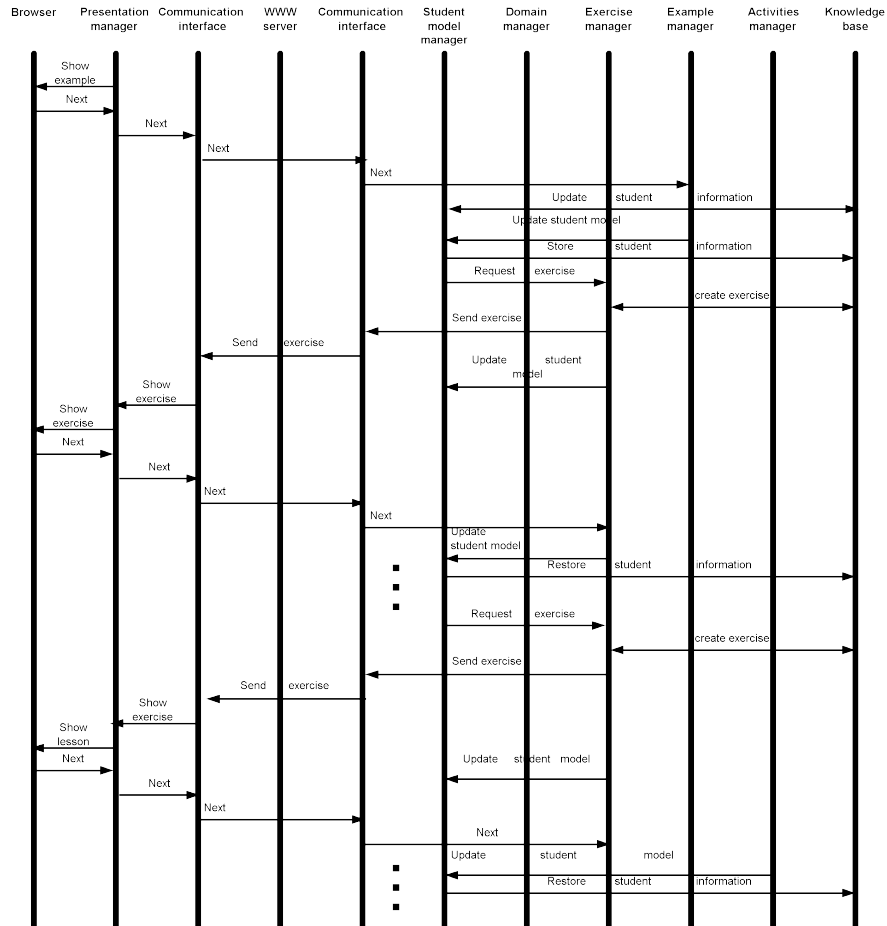


FIGURA 12 – Fluxo de Mensagens na Fase dos Exercícios.

A fase de exercícios, mostrada na figura 12 ocorre de modo semelhante: sempre que o *Student model manager* julgar conveniente, normalmente após uma seqüência de exemplos, ele invoca o *Exercise manager* para criar e enviar ao aluno uma seqüência de exercícios adequada ao seu nível de conhecimento e ao estado em que se encontra naquele momento.

Da mesma forma que a situação das lições e dos exemplos, os exercícios são enviados ao *Presentation manager* através dos agentes denominados *Communication interface*. Os exercícios, no entanto têm um grau de interação maior com o aluno e a atuação do agente *Exercise manager* é mais complexa na medida em que ele é um dos principais responsáveis por avaliar o grau de entendimento do aluno naquela lição e

informar ao *Student model manager* que dá andamento ao processo de coordenação das ações de acordo com o resultado da avaliação feita.

Neste momento, o *Student model manager* pode determinar uma nova seqüência de exercícios com um grau de dificuldade diferente do anterior, para avaliar melhor a situação ou para levar o aluno a uma reflexão mais profunda sobre a matéria, pode solicitar ajuda ao *Example manager* para que atue no sentido de dar novos exemplos ao aluno na tentativa de que ele possa aprofundar o seu entendimento ou, ainda, invocar o *Domain manager* para retomar algum ponto que o *Student model manager* julgue não ter ficado claro para o aluno.

Embora o *Student model manager* detenha o controle global sobre o processo, o aluno pode interferir no andamento da sessão solicitando alguma ajuda em particular, provocando a invocação do respectivo agente para prestar o auxílio solicitado.

A implementação das mensagens [BIC 99], ilustrada na figura 13, foi feita através de dois módulos: um formato geral que possui especificação KQML (classe *KQMLMessage*), e um formato específico que compõe a tarefa que o agente deve executar (classe *ContentMessage*). A especificação da mensagem foi elaborada desta maneira pois o RMI permite a passagem de objetos criados pelo implementador.

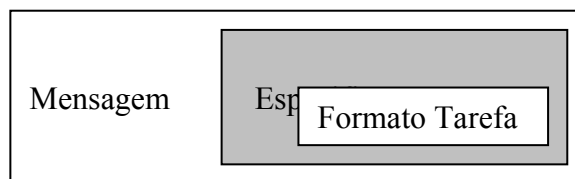


FIGURA 13 – Módulos da Mensagem

A linguagem KQML possui um conjunto de parâmetros-chave reservados, sendo útil para estabelecer uma uniformidade dos parâmetros e suporte para que programas entendam *performatives* desconhecidas, mas com parâmetros conhecidos. Os parâmetros reservados e seus significados estão demonstrados na TABELA 6, tais parâmetros foram implementados na forma de uma classe Java, chamada *KQMLMessage*, como mostra a figura 14.

TABELA 6 – Parâmetros-Chave das *Performatives*

Parâmetros-Chave	Significado
<i>:sender</i>	o atual transmissor de uma <i>performative</i>
<i>:receiver</i>	o atual receptor da <i>performative</i>
<i>:language</i>	o nome da linguagem de representação do parâmetro <i>:content</i>
<i>:ontology</i>	o nome da ontologia usada no <i>content</i>
<i>:in-reply-to</i>	o nome esperado para resposta
<i>:reply-with</i>	se o transmissor espera uma resposta e, se for sim, um rótulo para resposta
<i>:force</i>	se o transmissor irá sempre negar o significado da <i>performative</i>
<i>:content</i>	a informação sobre o que a <i>performative</i> expressa

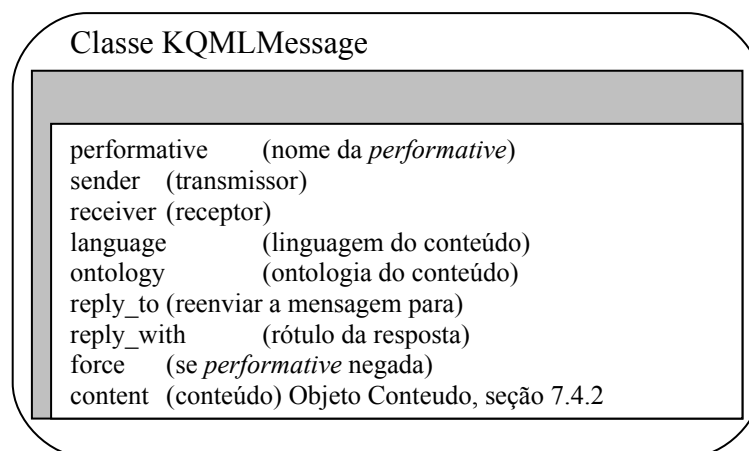


FIGURA 14 – Classe KQMLMessage

A TABELA 7 identifica as *performatives* KQML [DAR 93], ‘S’ significa o remetente e ‘R’ o receptor.

TABELA 7 - *Performatives* Utilizadas pelos Agentes

Nome	Resposta requerida	Significado
<i>Achieve</i>		S deseja que R torne algo de verdadeiro em seu ambiente
<i>Break</i>		S deseja que R rompa um caminho estabelecido
<i>Delete_all</i>		S deseja que R remove todas as sentenças que se encontram na BCV (Base de Conhecimento Virtual)
<i>Delete_one</i>		S deseja que R remove uma sentença da BCV
<i>Error</i>		S considera uma mensagem de R mal formulada
<i>Forward</i>		S deseja que R repasse a mensagem para outro receptor
<i>Insert</i>		S pergunta a R para adicionar o conteúdo (:content) na BCV
<i>Sorry</i>		S não pode mais prover informações para o <i>reply</i> da <i>performative</i>
<i>Tell</i>	X	a sentença está no VKB do S

Seguindo o formato de mensagem KQML, é necessário especificar o parâmetro *content*, para isso, foi implementada a classe *ContentMessage*, como mostra a figura 15, que possui todos os parâmetros necessários para compor uma tarefa. Tais parâmetros são utilizados conforme o agente e o tipo de mensagem que este deseja enviar.

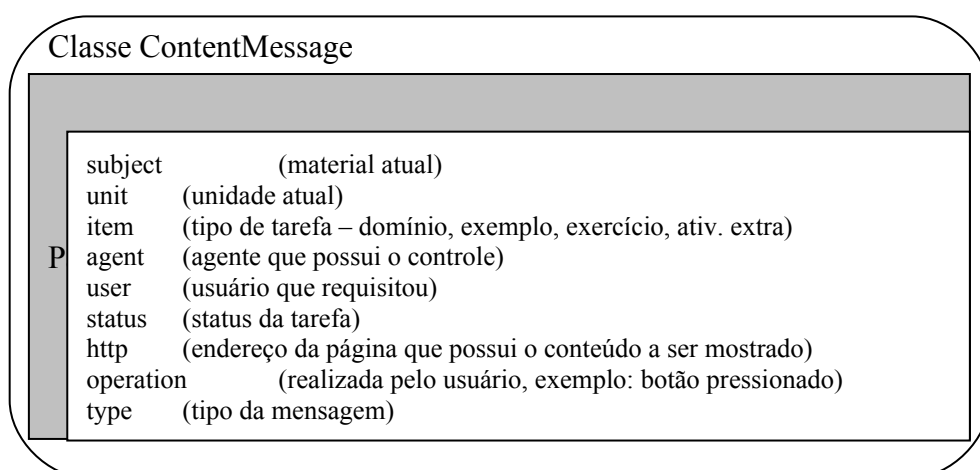


FIGURA 15 – Classe ContentMessage

Os tipos de conteúdo foram mapeados na TABELA 8. De acordo com o conteúdo, diferentes parâmetros são utilizados na mensagem.

TABELA 8 – Conteúdo das Mensagens

Conteúdo das Mensagens		
Agente	Conteúdo de Entrada	Conteúdo de Saída
Gerenciador de conteúdo	RECEBENDO TAREFA <ul style="list-style-type: none"> • Unidade , • item da Unidade , • Usuário que solicitou, • Agente que solicitou (se existir) 	ENVIANDO UMA TAREFA AO USUÁRIO <ul style="list-style-type: none"> • Conteúdo do item da Unidade (http://...), • Usuário que solicitou, • Agente que solicitou (se existir) ENVIANDO FIM DE TAREFA AO AGENTE 5 <ul style="list-style-type: none"> • Agente, • Usuário, • Status da tarefa
Gerenciador de exercícios	RECEBENDO TAREFA <ul style="list-style-type: none"> • Unidade , • exercício da Unidade, • Usuário que solicitou, • Agente que solicitou (se existir) 	ENVIANDO UMA TAREFA AO USUÁRIO <ul style="list-style-type: none"> • Conteúdo do exercício da Unidade (http://...), • Usuário que solicitou, • Agente que solicitou (se existir) ENVIANDO FIM DE TAREFA AO AGENTE 5 <ul style="list-style-type: none"> • Agente, • Usuário, • Status da tarefa
Gerenciador de exemplos	RECEBENDO TAREFA <ul style="list-style-type: none"> • Unidade , • Exemplo da Unidade, • Usuário que solicitou, • Agente que solicitou (se existir) 	ENVIANDO UMA TAREFA AO USUÁRIO <ul style="list-style-type: none"> • Conteúdo do exemplo da Unidade (http://...), • Usuário que solicitou, • Agente que solicitou (se existir) ENVIANDO FIM DE TAREFA AO AGENTE 5 <ul style="list-style-type: none"> • Agente, • Usuário, • Status da tarefa
Gerenciador de interação	RECEBENDO TAREFA <ul style="list-style-type: none"> • Unidade , • Atividade Extra da Unidade, • Usuário que solicitou, • Agente que solicitou (se existir) 	ENVIANDO UMA TAREFA AO USUÁRIO <ul style="list-style-type: none"> • Atividade Extra da Unidade (animação, VRML , etc...), • Usuário que solicitou, • Agente que solicitou (se existir) ENVIANDO FIM DE TAREFA AO AGENTE 5 <ul style="list-style-type: none"> • Agente, • Usuário, • Status da tarefa

Conteúdo das Mensagens		
Agente	Conteúdo de Entrada	Conteúdo de Saída
Gerenciador do modelo do aluno	<p>RECEBENDO FIM DE TAREFA DO AGENTE PARA DETERMINADO USUÁRIO</p> <ul style="list-style-type: none"> • Agente, • Usuário, • Status da tarefa (fim da unidade, exemplos, exercícios, atividades extras – em que pé ficou, dados do aluno) <p>RECEBENDO LOGIN USUÁRIO</p> <ul style="list-style-type: none"> • Identificação do Usuário e Senha 	<p>ENVIANDO A TAREFA AO AGENTE QUE TOMARÁ O CONTROLE</p> <ul style="list-style-type: none"> • Agente, • Usuário, • Unidade, • Tarefa :exemplo ou :exercício ou :domínio ou :atividade extra
Gerenciador da interface	<p>RECEBENDO CONTEÚDO</p> <ul style="list-style-type: none"> • Agente, • Usuário, • Conteúdo a ser mostrado (domínio, exercício, exemplo, atividade extra) 	<p>ENVIANDO SOLICITAÇÃO DO USUÁRIO</p> <ul style="list-style-type: none"> • Agente, • Usuário, • Solicita (Botão pressionado – próximo exercício, volta/próxima atividade extra, volta/próxima página domínio), etc...

A figura 16 ilustra uma mensagem enviada pelo Agente Interface ao agente Gerenciador Modelo do Aluno com o conteúdo *Login do Usuário*. Para enviar uma mensagem é necessário criar dois objetos, respectivamente, herdados das classes KQMLMessage e ContentMessage.

```
KQMLMessage mensagem = new KQMLMessage();
ContentMessage conteúdo = new ContentMessage();
```

O objeto conteúdo será encapsulado pelo parâmetro *content* do objeto mensagem, que por sua vez será transmitido ao agente especificado pelo parâmetro *receiver*.

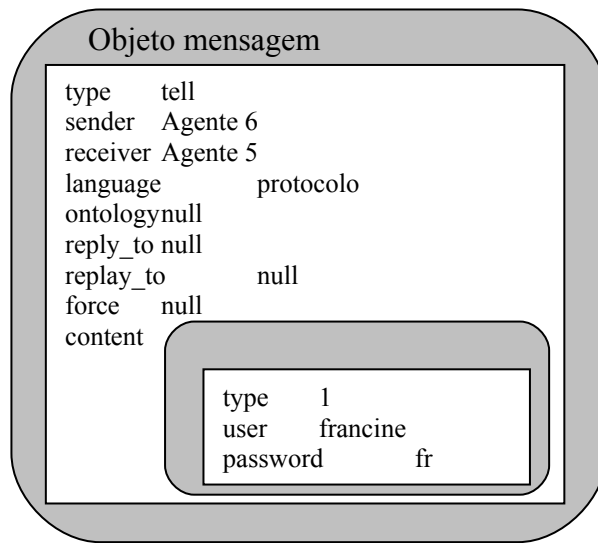


FIGURA 16 – Mensagem com Conteúdo *Login do Usuário*

3.5 Modelo de agente

Um dos objetivos deste trabalho é, além da discussão de um modelo organizacional de uma sociedade de agentes que em conjunto constitua-se em uma infra-estrutura de desenvolvimento de ambientes de ensino inteligentes, proponha também um modelo interno dos agentes a fim de viabilizar a sua implementação. É também objetivo deste trabalho propor avanços na modelagem de estados mentais internos dos agentes. Estes pontos serão apresentados a seguir.

A fim de poder atuar sobre o ambiente, cada agente possui uma representação interna parcial do mundo que o rodeia. Utilizando para isso, a metáfora de *estados mentais* para modelar a base de conhecimentos que representam os estados do ambiente onde o agente está inserido.

Diversos trabalhos [SHO 93], [THO 93], [RAO 95], [RAO96], [MÜL 96] descrevem os conceitos referentes a estes estados e outros componentes de agentes cognitivos, utilizados neste projeto, os quais resumem-se a seguir (os termos originais em inglês foram colocados entre parênteses para uma melhor identificação com o termo empregado originalmente na literatura referida):

- **Crenças** (*beliefs*): contém as visões fundamentais de um agente sobre seu ambiente. Representam o estado corrente do mundo interno e externo do agente, e é atualizado sempre que alguma nova informação pertinente é recebida. Um agente pode possuir crenças sobre o seu ambiente, sobre as crenças de outros agentes, sobre interações com outros agentes e crenças sobre as suas próprias crenças.
- **Desejos** (*desires*): são derivados diretamente das crenças, e correspondem à avaliação do agente a respeito de situações futuras. Representam crenças que constituem estados desejados, independentemente da sua possibilidade de realização.
- **Objetivos** (*goals*): Representam o conjunto de desejos sobre o qual o agente pode atuar. Em contraste com os desejos, os objetivos são realísticos e não devem conflitar entre si.

- **Intenções** (*intentions*): é um conjunto de objetivos que constituem acordos usualmente estabelecidos entre agentes em atingir um determinado estado de crenças do mundo em um determinado momento. Como um agente pode não possuir todos os recursos necessários para atingir seus objetivos, eles devem ser priorizados, de acordo com determinadas regras. As intenções são similares aos comprometimentos, no que se referem à concordância por parte de um agente em executar ações em nome de outro, mas difere-se no sentido de que, em vez de uma única ação uma intenção é uma concordância em executar as ações necessárias para atingir um estado-objetivo do meio ambiente. Uma requisição por parte de um agente, em termos de um objetivo de alto nível, permite que o agente que receba a solicitação e aceite, tenha a liberdade de alcançá-la usando os meios que lhe parecerem mais adequados.
- **Planos** (*plans*): são combinações de intenções em unidades consistentes.
- **Comprometimentos** (*commitments*): um comprometimento é uma concordância por parte do agente em executar uma ação em particular em um determinado momento. Quando um agente solicita uma ação a outro, este aceita o compromisso e executa a ação no momento adequado, se as pré-condições estiverem estabelecidas.
- **Ações** (*actions*): são interferências que um agente faz no seu ambiente interno e externo. Podem ser divididas em três categorias: ações de comunicações, que referem-se ao envio de mensagens, ações privativas, que são métodos que o agente utiliza para cumprir tarefas, tais como acessar uma base de dados, acionar um objeto na interface de usuário, etc., e ações internas que consistem na revisão de suas crenças.
- **Aptidões** (*capabilities*): são constructos utilizados pelo agente para associar uma ação com as pré-condições necessárias a sua execução. Uma lista de aptidões de um agente define as ações que o agente pode executar se as pré-condições forem satisfeitas. Usualmente esta lista é estática e permanece durante o tempo de vida de um agente, embora as ações que ele possa executar variem de acordo com as condições existentes. As aptidões de um agente são utilizadas para decidir se um agente pode, ou não, adotar um comprometimento em executar uma ação.
- **Eventos** (*events*): São ações percebidas no ambiente onde se insere o agente e que foram realizadas por outro agente, humano ou programa. Pode constituir se em uma mensagem recebida, uma ação de um usuário através de uma interface, etc..
- **Regras de comportamento** (*behavioral rules*): determina o curso de ação do agente a cada momento. Estas regras casam com o conjunto de possíveis respostas do estado corrente do ambiente, de acordo com as crenças que o agente possui a respeito dele. Regras de comportamento podem ser vistas como cláusulas do tipo QUANDO – SE – ENTÃO, ou seja, QUANDO um novo evento ocorre no ambiente do agente, e SE as crenças sobre o estado corrente correspondem às condições necessárias, ENTÃO são executadas as ações.

Para atuar em uma sociedade de agentes com a organização que foi discutida anteriormente e para desempenhar os papéis que estão definidos naquela organização social, o agente deve possuir algumas características fundamentais:

- Deve possuir uma representação interna do ambiente onde se insere para suportar as transações descritas através de troca de mensagens com um conteúdo complexo;
- Deve possuir mecanismos de troca de informações com os demais agentes que suportem o tipo de mensagem adequado para as informações que devem ser trocadas;

- Deve possuir habilidades para desempenhar o papel específico que lhe foi designado, e que estas habilidades possam ser implementadas sem alterar substancialmente a natureza básica do agente;

Diversas alternativas foram examinadas na literatura para a definição de uma arquitetura adequada para implementar os agentes definidos no modelo. Em especial foram avaliados três instrumentos de desenvolvimento de sistemas multiagentes já discutidos no capítulo 2.3: o *Agent Builder* [RET 99], um conjunto integrado completo, desenvolvido pela *Reticular Systems* para construção de sistemas multiagentes, em Java, baseado na construção de agentes com arquitetura BDI, proposta por [SHO 93]; o *JAFMAS*, que propõe uma metodologia e provê uma estrutura para desenvolvimento de sistemas multiagentes, oferecendo uma biblioteca de classes em Java que permite a implementação de sistemas com ênfase na comunicação entre agentes através da linguagem KQML [DEE 98]; e o *JATLite*, [STA 99] desenvolvida pelo *Computer Science Department at Stanford University*, que provê um conjunto de *Java Packages* que constitui uma infra-estrutura para o desenvolvimento de sistemas multiagentes em linguagem Java. Os três sistemas provêm uma estrutura básica de comunicação que permite a troca de mensagens KQML entre agentes.

Baseado nisso, foi definido um modelo teórico inicial, baseado na abordagem de Orientação a Objetos, a partir do qual as diversas implementações feitas ao longo do desenvolvimento do projeto se basearam. A figura 17 mostra a estrutura deste modelo.

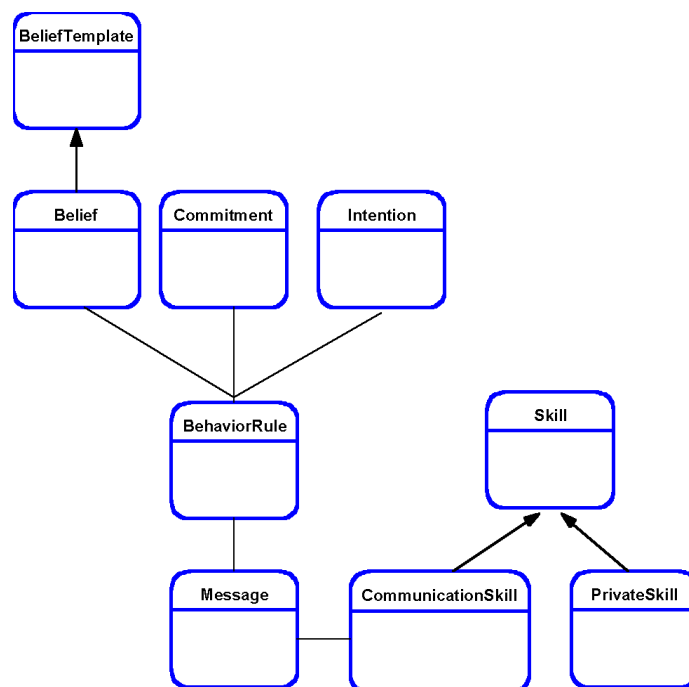


FIGURA 17 – Modelo Interno de Agente.

O princípio utilizado foi o emprego de classes cujas instâncias representassem os elementos que constituem os estados mentais do agente. Deste modo, cada um dos conjuntos de elementos descritos anteriormente (crenças, comprometimentos, etc.) é representado por uma classe de objetos.

O ciclo de execução do agente consiste dos seguintes passos:

- processamento de novas mensagens: neste momento, o agente decompõe a mensagem e reconhece qual a tarefa que esta propõe;
- determinação da regra de comportamento: a partir da análise da mensagem, o agente determina qual regra de comportamento é aplicável para a corrente situação e, se necessário, delega a tarefa a outro(s) agente(s);
- execução de ações pessoais e de comunicação: cada regra de comportamento possui ações pessoais e de comunicação, e é neste momento que tais ações são executadas pelo agente;
- atualização do estado mental: as ações executadas pelo agente no nível anterior podem alterar o seu conhecimento a respeito do mundo, então, é necessário que ocorra um gerenciamento do seu conhecimento a respeito do mundo.

A figura 18, a seguir, ilustra este processo.

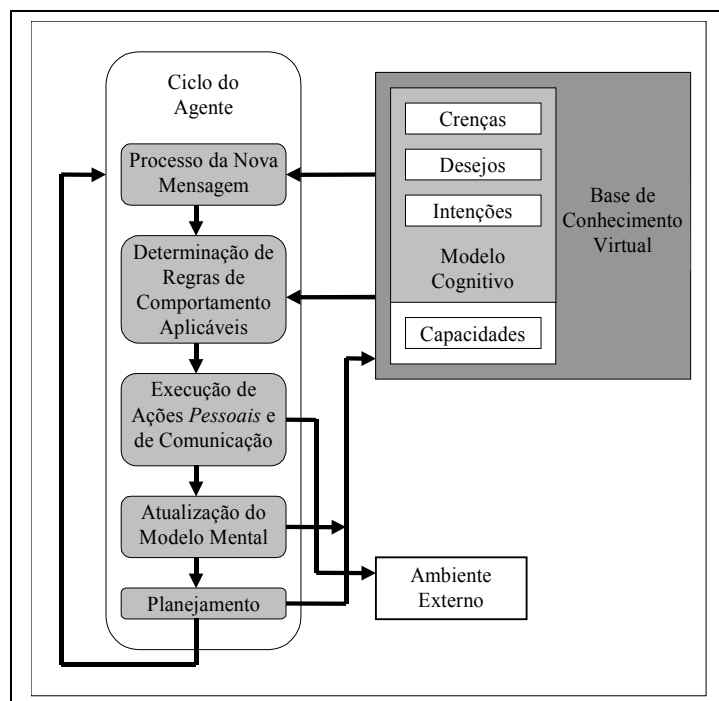


FIGURA 18 - Processo de Execução do Agente Genérico

3.6 Implementação

De acordo com os objetivos estabelecidos neste trabalho, a implementação do JADE e a utilização dele na construção de um ambiente de ensino foram contemplados. Neste item, serão discutidos alguns aspectos da implementação de JADE e, posteriormente, será apresentado o protótipo Eletrotutor.

Um dos primeiros aspectos na implementação diz respeito à tecnologia e à linguagem de programação utilizada. Além das linguagens tradicionais de programação, como C e C++, algumas linguagens têm se destacado na construção de sistemas

multiagentes. Outras têm sido desenvolvidas com este propósito específico. [WAL 98] faz uma análise das principais linguagens que têm sido utilizadas com este propósito.

Segundo este autor, os requerimentos desejados em uma linguagem orientada a agentes são:

- orientação a objeto;
- independência de plataforma;
- capacidade de comunicação;
- segurança;
- manipulação de código em tempo de execução;
- reatividade;
- suporte à multitarefa;
- armazenamento de dados persistentes;
- extensibilidade.

Entre as linguagens apontadas por [WAL](Actors, Agent-0, Tcl/TK, Safe-Tcl, Agent-tcl, Java, Telescript, Active *Web* Tools, Python, Oblic, April, Scheme-48 e RTA/ABLE) aquela que melhor atende a estes requisitos e nos pareceu a mais adequada para os objetivos desta proposta é a linguagem Java.

A linguagem Java, desenvolvida pela Sun Microsystems [SUN 98] é baseada em conceitos que são particularmente adequados para o desenvolvimento de agentes inteligentes, entre eles, o conceito de linguagem fortemente orientada a redes e à independência de plataforma.

Em linguagens tradicionais de programação, um compilador ou um interpretador é utilizado para converter o código-fonte do programa em um código binário específico para a plataforma na qual o programa será executado. Se um mesmo programa ou diferentes partes de um programa tiverem que ser executados em plataformas diferentes, cada uma destas partes deverá ser compilada da maneira adequada. O compilador da linguagem Java, por sua vez, converte os programas-fonte em *byte-code* que são códigos semicompilados que executam em todas as plataformas que suportam Java, por meio de um interpretador adequado.

O *byte-code* pode ser executado localmente como uma aplicação ou transferido através da rede para um computador remoto, onde será executado como um *Java applet*. Através da WWW, o *Java applet* é embutido em uma página HTML e executada no computador remoto através de um navegador (Browser) comercial que suporte a linguagem através do ambiente de execução. Os navegadores comerciais mais populares vêm com o suporte à linguagem, propiciando o ambiente adequado à execução do programa.

Além das características citadas, a linguagem Java tem as facilidades de ser uma linguagem orientada a objetos, bastante semelhante à C++, possui inúmeras facilidades de suporte a redes tais como a implementação de *sockets* através de protocolo TCP/IP, *multicast sockets* UDP, que permite a implementação de difusão de mensagens (*broadcasting*), e os mecanismos denominados *Object Serialization* e *Remote Method Invocation* (RMI) que permitem o envio de estruturas complexas de objetos e a execução de métodos disparados remotamente. Além do mais, a linguagem Java vem evoluindo bastante, incorporando novos conceitos como o uso dos chamados *Servlets*, bem como integração com novos padrões e novas linguagens como Corba e XML.

A incorporação de algumas destas evoluções da tecnologia da linguagem Java na versão mais recente do protótipo Eletrotutor, demonstrou que a arquitetura do JADE é facilmente aderente a inovações tecnológicas sem alterações na sua estrutura.

A fim de atender algumas motivações importantes desta proposta, que se referem à premissa de desenvolver uma arquitetura bastante evolutiva, com significativo aproveitamento de código e que incorpore ao máximo o uso de ferramentas e metodologias consagradas de desenvolvimento de projetos, foram levantados que tipos de ferramentas poderiam ser adotadas no desenvolvimento do JADE. Três tipos de ferramentas foram utilizadas: uma, denominada VISIO, empregada na elaboração dos modelos de objeto, diagramas de fluxo de dados e de funções; outra, denominada KAWA para servir de modelo para a implementação dos agentes; e uma terceira, do tipo IDE (*Interface Development Environment*) que empresta produtividade e facilidade de documentação nos códigos escritos. Durante etapa de engenharia de software, foi utilizada também a ferramenta *Together J++*, produto comercializado pela empresa *Object International Inc.*, [OII 99]

3.6.1 A arquitetura global do JADE

A arquitetura global do JADE é então constituída pelos agentes autônomos já descritos anteriormente, e que possuem, conforme foi visto, mecanismos de cooperação entre si. Cada agente possui as habilidades que lhes permitem interagir com os demais agentes que fazem parte da comunidade onde está inserido e com os alunos, através do Gerenciador de Comunicação, e da Interface do Aluno.

O código do programa que trata da interface do aluno é enviado ao ambiente do usuário, constituído pelo *browser* na forma de um *Java applet*. Ele é responsável pela montagem das telas de interface para o aluno, pela interação homem-máquina com o mesmo e pela troca de mensagens com os demais agentes, juntamente com o Gerenciador de Comunicação.

A necessidade do agente de comunicação deve-se basicamente às limitações impostas pelos mecanismos de segurança inerentes à linguagem Java, utilizada no JADE, que permitem que o *applet* enviado ao navegador, através do servidor WWW, se comunique somente com a máquina de onde ele se originou. Assim, para que o ambiente do aluno pudesse comunicar-se diretamente com cada agente pedagógico, todos eles deveriam estar fisicamente na mesma máquina, o que contraria as premissas estabelecidas anteriormente.

A conexão do aluno com o sistema pode ser feita por meio de linha telefônica comutada, através de um provedor comercial de acesso Internet, por exemplo, ou por uma rede local de uma escola que esteja de algum modo conectada a Internet. Os outros agentes comunicam-se entre si por meio de rede local ou remota, utilizando também os recursos da Internet.

Todos os agentes possuem mecanismos de comunicação para interagir entre si e, cada um deles, possui habilidades especializadas para desempenhar as funções para a qual foram projetados, como por exemplo, propor exercícios aos alunos e avaliar as suas respostas dando-lhes a realimentação adequada, gerar exemplos adequados ao estado cognitivo de cada aluno no presente momento, apresentar material instrucional da forma e no momento adequado, e assim por diante.

A figura 19 representa a arquitetura elaborada e os métodos de comunicação utilizados por eles.

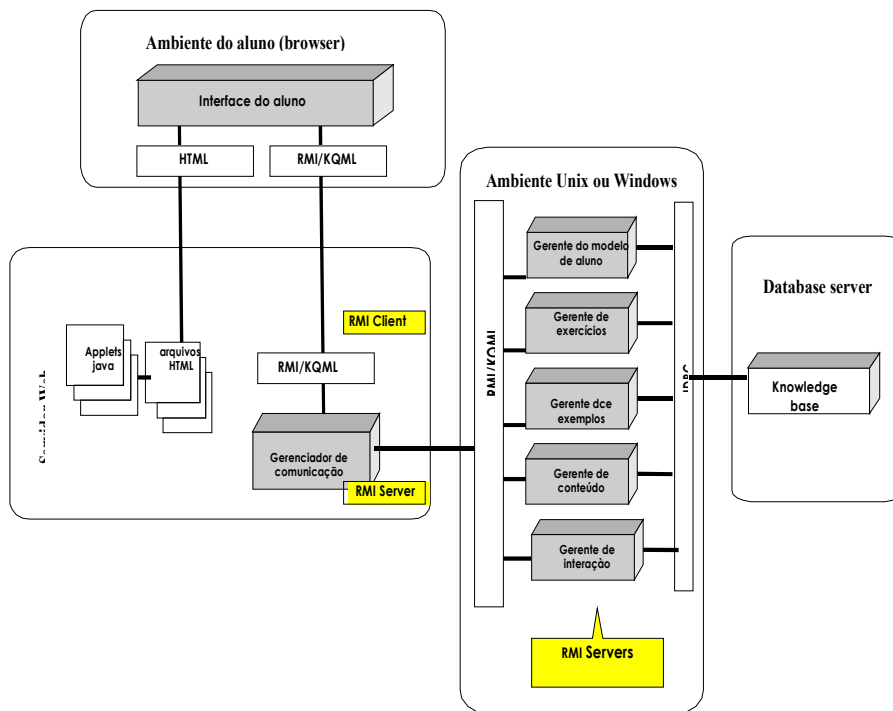


FIGURA 19 – Arquitetura do Ambiente

Os agentes do JADE necessitam, para o seu funcionamento, de um servidor de banco de dados relacional, de um servidor HTML (*Web server*) e de um ou mais ambiente operacional onde deverão ser hospedados os agentes. O sistema operacional pode ser qualquer um que suporte linguagem Java (*Unix, Windows 98, Windows NT, MacOS, etc.*).

Conforme foi visto, os agentes foram implementados como programas totalmente independentes e cada um deles resulta em um processo computacional independente, rodando sob o sistema operacional. Todo o sistema é, portanto, independente de plataforma, devendo, no entanto ser instalado, em cada máquina hospedeira dos agentes, o ambiente de *run time* da linguagem Java.

Todos os recursos, o banco de dados, o servidor HTML e os agentes podem, por isso, ser instalados em um único computador ou serem distribuídos em diferentes máquinas. A única limitação é que o agente responsável pela gerência de comunicação entre o ambiente dos alunos e os demais agentes deve ser instalado no mesmo local onde fica instalado o servidor HTML. Isto deve-se às restrições de segurança da linguagem Java que só permite que um *applet* comunique-se com o servidor de onde ele foi carregado.

O servidor de banco de dados pode ser qualquer banco de dados comercial que possua drivers de conexão compatíveis com Java (JDBC) para isso, algumas pequenas adaptações devem ser feitas no código, dependendo do banco utilizado. Neste trabalho foram feitos testes, todos bem-sucedidos, com o banco de dados *Oracle, SQL Server* e *MySQL*, utilizando diferentes classes Java de JDBC.

3.7 Desenvolvendo aplicações com o JADE

Um dos objetivos deste trabalho, conforme foi visto, é estabelecer ou definir uma metodologia de ensino adequada e compatível com os recursos propiciados pela infra-estrutura apresentada. Esta metodologia é, na realidade definida através das estratégias de ensino criadas pelo autor do curso durante a fase de desenvolvimento de cada aplicação, obviamente limitada pelos recursos e técnicas proporcionados, conforme apresentado no item 3.2. Estes aspectos serão discutidos aqui novamente e, mais adiante no capítulo referente à implementação do protótipo.

Para implementar um curso de Educação a Distância utilizando a infra-estrutura do JADE, o autor não necessita, a princípio conhecer nenhuma linguagem de programação. Todo o material instrucional pode ser elaborado utilizando-se um editor de texto simples e um editor de páginas HTML do tipo WYSIWYG (*What You See Is What You Get*) tal como o *Netscape Composer* que acompanha o pacote *Netscape*, ou mesmo o *Word* da *Microsoft*.

Com essas ferramentas, o autor consegue produzir material instrucional com relativa facilidade, embora o resultado seja algo simples. Caso deseje utilizar todos os modernos recursos disponíveis que podem ser inseridos nos arquivos HTML, como animações, clipes de áudio e vídeo, etc., ferramentas mais poderosas e algum conhecimento mais elaborado em produção de material para páginas *Web* é requerido.

Ainda assim, não é necessário nenhum conhecimento de programação para obter do JADE a adaptabilidade do ambiente, e o autor pode criar um curso estruturado, com metodologia robusta, acompanhando os passos descritos a seguir:

- Modelagem do curso: a primeira etapa envolve os procedimentos normais utilizados na elaboração de qualquer curso ou disciplina. Devem ser estabelecidos os objetivos pedagógicos do curso, as atividades a serem desenvolvidas para alcançar os objetivos traçados, a forma e os critérios de avaliação em cada etapa do curso.
- Elaboração dos conteúdos: nesta etapa deve ser reunido todo o material que o autor julgar conveniente para o aluno utilizar. Textos, figuras, gráficos, material audiovisual, etc. Como a idéia da metodologia empregada no JADE implica em riqueza de técnicas que podem ser oferecidas ao aluno de acordo com o seu perfil, quanto mais material for elaborado sobre um mesmo conteúdo, usando diferentes alternativas de técnicas de apresentação, mais adaptativo torna-se o ambiente do curso a ser desenvolvido. A limitação tecnológica do material é dada pela capacidade de veiculação na rede por meio de arquivos em linguagem HTML. O material deve ser produzido e reunido em diferentes seqüências de páginas HTML que irão, cada uma delas, formar uma técnica diferente de abordar cada conteúdo. Posteriormente, deverá ser informado ao sistema cada uma destas seqüências, inserindo no banco de dados a organização do curso e as páginas que formam as diversas seqüências de cada unidade.
- Elaboração dos exemplos: o JADE oferece ao autor uma série de exemplos interativos previamente elaborados, desenvolvidos em forma de *Java applets* que são manuseados pelos agentes de forma a obter informações a respeito do aluno durante a sua utilização. Os exemplos são gerados dinamicamente a partir de parâmetros que o *applet* obtém da página HTML onde ele está

inserido. Por isso, para a elaboração dos exemplos, o autor deverá inserir nas páginas HTML os parâmetros adequados, utilizando um editor de texto qualquer, com os textos dos exemplos e algumas diretrizes, que farão com que o sistema gere os exemplos de acordo com estes parâmetros.

- **Elaboração dos exercícios:** Na elaboração dos exercícios o autor deve seguir procedimentos semelhantes aos adotados na elaboração dos exemplos. Assim como os exemplos, o JADE oferece determinados tipos de exercícios previamente elaborados que são gerados dinamicamente a partir de parâmetros que o autor insere em locais determinados no texto da página HTML que transporta o *Java applet* que, por sua vez, gera o exercício, através de um editor de textos. Os mesmos critérios adotados na elaboração do material instrucional podem ser empregados na elaboração dos exemplos e dos exercícios, isto é, o autor pode criar um número indefinido de diversos tipos de exemplos e exercícios, com diferentes graus de dificuldade e, na modelagem do curso, o autor informa ao sistema as diferentes seqüências possíveis de exemplos e exercícios que os agentes podem manusear, em cada unidade de ensino, de forma a obter um método de ensino dinâmico e adaptativo como convém a um sistema computacional inteligente de ensino.
- **Definição das técnicas de ensino:** depois de produzir um repertório de material instrucional, assim como uma diversidade de exemplos e exercícios que contemple os objetivos do curso, com uma variada gama de opções, o autor deve organizar este material de modo a obter diferentes combinações de formas de apresentação de conteúdos, de exemplos e de exercícios separadamente e organizá-las de modo conveniente, definindo assim diferentes técnicas de ensino a serem aplicadas a diferentes perfis de alunos.
- **Construção da base de conhecimentos do sistema:** definido o projeto pedagógico do curso, o autor deve informar ao sistema a estrutura do mesmo, alimentando a base de dados. Para esta finalidade foi implementada [REI 2000] uma ferramenta administrativa que permite inserir ou modificar as informações necessárias no banco de dados que forma a base de conhecimentos do sistema. Com ela o autor insere os links onde se encontram armazenados os componentes do material didático utilizado em cada tática de ensino, a correta associação entre as diversas estratégias de ensino com as suas respectivas táticas, bem como as regras de avaliação e de andamento das lições.

O JADE, como se pode perceber, ainda carece de ferramentas que tornem o trabalho de autoria mais fácil de ser realizado por técnicos do conteúdo a ser desenvolvido, embora importantes avanços tenham sido obtidos. No entanto, acredita-se que um projeto consistente de desenvolvimento de ambiente de ensino auxiliado por computador não pode dispensar nenhuma das três principais áreas envolvidas: o responsável pelo projeto pedagógico, o responsável pelo conteúdo e o responsável pela implementação do sistema. Mesmo assim, avanços ainda maiores poderiam ser obtidos com o desenvolvimento de uma ferramenta de autoria que orientasse todo o processo.

4 O protótipo Eletrotutor

Neste capítulo será descrito o protótipo que vem sendo utilizado durante o desenvolvimento do JADE, o Eletrotutor. Conforme já foi citado, o Eletrotutor aborda algumas unidades da Física, mais precisamente da Eletricidade, abrangendo Eletrodinâmica e a Lei de Ohm. Sua primeira versão foi implementada como parte da dissertação de mestrado apresentada em [SIL 92] e [SIL94]. Posteriormente, uma segunda versão foi implementada [SIL 96b], também com objetivos de avaliação de efetividade de sua arquitetura como ferramenta educacional, porém, sob outro paradigma.

A primeira versão foi implementada em linguagem ARITY-PROLOG. De acordo com a arquitetura tradicional de sistemas tutores inteligentes. Já a segunda versão foi desenvolvida em linguagem HTML e *JavaScript*, com uma interface gráfica, de acordo com uma arquitetura de tutores convencionais usualmente utilizada em ambientes de ensino via Internet.

A terceira versão do Eletrotutor foi desenvolvida por [BIC 99], utilizando e aperfeiçoando os conceitos e as definições de arquitetura que tinham sido estabelecidos até então para o JADE, conforme será descrito adiante. Posteriormente, esta versão foi aperfeiçoada incorporando a contribuição de [ASC 2000] nos aspectos referentes ao projeto da interface e, finalmente, uma quarta versão está sendo implementada por [REI 2000] com os últimos avanços, e outros a serem incorporados futuramente na arquitetura do JADE.

O conteúdo do Eletrotutor é constituído por alguns capítulos de Eletrodinâmica, uma área da Física que estuda alguns fenômenos da Eletricidade e aborda as relações entre algumas grandezas elétricas como Corrente Elétrica, Tensão, ou Diferença de Potencial, Resistência e Potência Elétrica.

A TABELA 9 apresenta em detalhes as unidades que constituem o Eletrotutor e os seus objetivos. As técnicas didáticas adotadas dividem-se em três: a apresentação de conteúdos teóricos, a apresentação de mecanismos de apresentação e manipulação, por parte do aluno, de exemplos práticos e a proposição de exercícios de diferentes tipos e em diferentes graus de dificuldade.

Os conteúdos são mostrados por meio de telas, que são apresentadas à medida que transcorre a matéria, através de um conjunto de informações denominadas quadros ou *frames*, constituídas por textos e figuras. Estas telas têm por objetivo apresentar a matéria ao aluno de diversas formas, e em uma seqüência que é montada dinamicamente junto com os exercícios e os exemplos até que o sistema conclua que o aluno compreendeu a lição atingindo o objetivo.

Os exemplos são constituídos por diversas formas de aplicação das equações que descrevem os fenômenos explicados, nas quais o aluno é convidado a fornecer valores a sua escolha, após, o tutor utiliza estes valores nas equações e as resolve, demonstrando a sua aplicação. O aluno pode repetir cada exemplo quantas vezes desejar, fornecendo, ao tutor, valores diferentes até que compreenda perfeitamente a descrição matemática do fenômeno, quando então pode passar, ou para o exemplo seguinte, ou para os exercícios da lição.

TABELA 9 – Características do Eletrotutor.

Projeto pedagógico do Eletrotutor		
População alvo	Alunos de Física do Segundo Ano do Segundo Grau	
Pré-requisitos	Matemática	Álgebra
		Equações de Primeiro Grau
		Múltiplos e Submúltiplos de Sistema de Medidas
		Arredondamento
	Física	Mecânica
Química	Tabela Periódica dos Elementos	
	Teoria Atômica	
Objetivos	O aluno, ao final do curso deve ser capaz de:	Descrever a Estrutura da Matéria.
		Conceituar Cargas Elétricas, Eletrização, Condutores e Isolantes.
		Conceituar Carga Elétrica Elementar e aplicar a Lei de Coulomb.
		Conceituar Campo Elétrico.
		Conceituar Potencial Elétrico e Diferença de Potencial.
		Conceituar Corrente Elétrica.
		Aplicar a Lei de Ohm.
Aplicar a Lei de Joule.		
Conteúdos	Teoria Atômica	
	Cargas Elétricas	
	Isolantes e Condutores	
	Lei de Coulomb	
	Campos Elétricos	
	Potencial Elétrico	
	Diferença de Potencial	
	Corrente Elétrica	
	Lei de Ohm	
	Resistência Elétrica	
	Circuitos Elétricos	
	Potência Elétrica	

Evidentemente, não foi objetivo deste trabalho a produção de um ambiente completo para utilização em sala de aula. Portanto, o desenvolvimento do material instrucional, dos exemplos e dos exercícios limitou-se ao mínimo necessário para que houvesse uma quantidade razoável de material que caracterizasse um protótipo a fim de verificar a validade das soluções da arquitetura proposta.

Na versão atual, existem exercícios referentes às questões teóricas, os quais são propostos na forma de frases tipo verdadeiro-ou-falso, onde a cada frase o aluno assinala V ou F para respondê-lo e exercícios que envolvem a aplicação de fórmulas.

Neste caso, para cada ponto da matéria, são apresentados vários tipos de problemas. Em cada tipo, são gerados de forma randômica os valores propostos nos exercícios, permitindo, assim, que o aluno possa fazer um número exaustivo de cada tipo de exercício, sempre com valores diferentes, e sempre com o tutor fornecendo a resposta correta, caso o aluno não acerte.

A partir da modelagem do curso anteriormente descrita e, definido um projeto pedagógico para o Eletrotutor, elaborou-se um Modelo Organizacional, um Modelo de Agente e um Modelo de Cooperação, que serviram de base para a elaboração dos modelos utilizados pelo JADE.

A seguir, serão descritos os principais aspectos que envolveram a implementação do Eletrotutor III. Para maiores detalhes, ver [BIC 98] e [BIC 99]

4.1 O Modelo Organizacional do Eletrotutor III

A arquitetura utilizada pelo Eletrotutor III foi a primeira implementação a utilizar, com algumas adaptações, o modelo organizacional do JADE. Assim, o ambiente Eletrotutor III é constituído por seis agentes autônomos, conforme mostra a figura 20

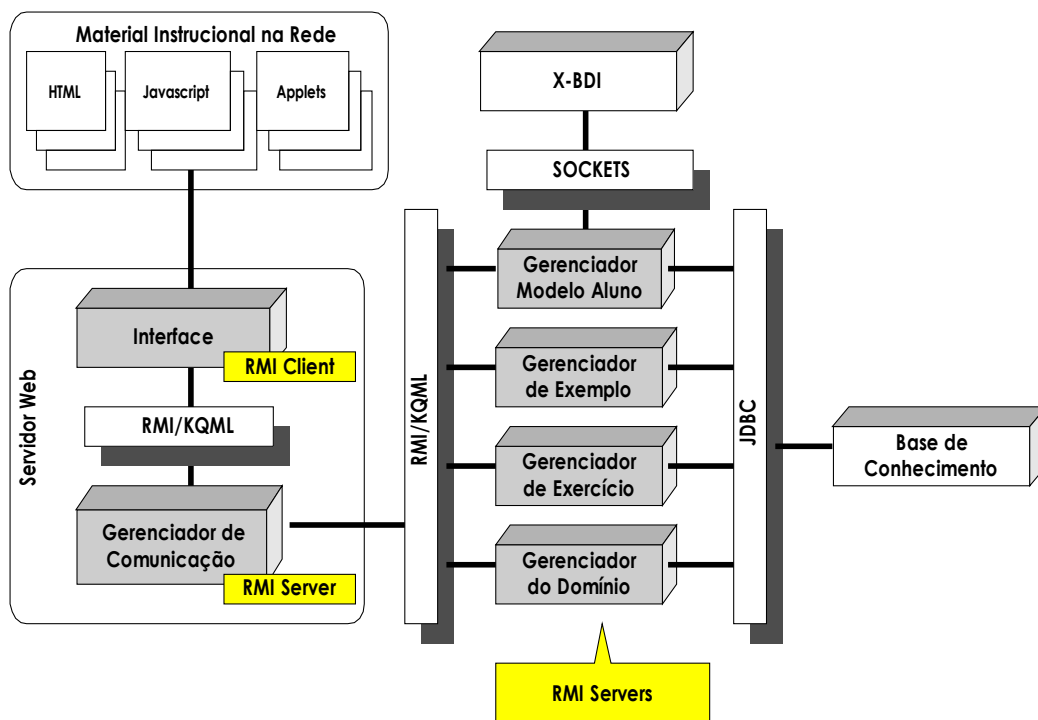


FIGURA 20 – Arquitetura do Ambiente Eletrotutor III

As características e as funções dos agentes do Eletrotutor são aquelas já descritas anteriormente no capítulo 3.3.

4.2 O modelo interno dos Agentes

O modelo interno dos agentes, utilizado nesta versão, apresentado por [BIC 99], trouxe avanços no modelo básico apresentado anteriormente, no capítulo 3, e foi baseado nos trabalhos de Shoham [SHO 91][SHO 93], Móra [MOR 97a][MOR 98] e Brenner [BRE 97]. O modelo, conforme ilustrado na figura 21, possui uma estrutura composta pelos seguintes elementos:

- primitivas de estados mentais: a fim de poder atuar sobre o ambiente, cada agente possui uma representação interna parcial do mundo que o cerca, empregando a metáfora de estados mentais (crenças, desejos e intenções de acordo com a arquitetura BDI) para modelar a base de conhecimento que representam os estados de cada agente em relação ao ambiente no qual encontra-se inserido e
- dois tipos de ação: as capacidades pessoais e de comunicação.

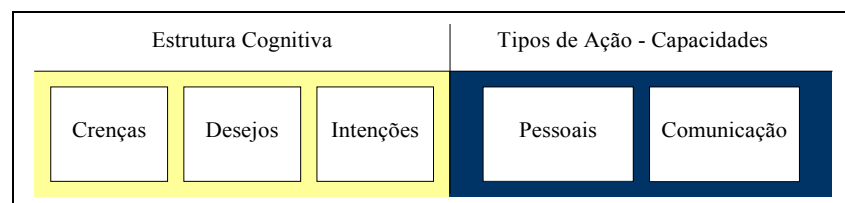


FIGURA 21 – Estrutura do Agente Genérico

A estrutura cognitiva contém os estados mentais que compõem o agente genérico e as regras que governam as interações entre tais estados mentais, e conseqüentemente, o comportamento do agente. Tal estrutura é constituída pelo conjunto de crenças, o conjunto de desejos e o conjunto das intenções do agente. Já, as capacidades do agente é o conjunto de habilidades que podem ser utilizadas pelo agente para associar uma ação no seu meio ambiente, de acordo com determinadas pré-condições, necessárias para realizar uma ação. As habilidades podem ser individuais, ou seja, próprias do agente, de acordo com a função que ele desempenha, e de comunicação, que são comuns a todos os agentes, independentes das suas funções.

A figura 22 representa o diagrama das principais classes implementadas [BIC 99] que compõem o ambiente Eletrotutor III, mostrando as relações **utiliza** e **implementa**. Segundo os conceitos da linguagem Java, para que uma classe implemente outra classe do tipo *interface* ela deve conter todos os métodos da classe *interface* implementados. As classes são as seguintes:

- **Agent**: classe que possui as especificações dos agentes;
- **KQMLMessage**: implementa o cabeçalho das mensagens;
- **ContentMessage**: implementa o conteúdo das mensagens;
- **server**: *interface* Java RMI;
- **Interface**: implementa o agente Interface, é o único RMI *client* do ambiente, representa a interface gráfica deste (*applet*);
- **ServerApplet**: implementa o RMI *server* que se comunica com o agente Interface (classe Interface.java) e com demais agentes do ambiente, equivale ao agente Gerenciador de Comunicação, deve ser executado na mesma máquina que o agente Interface;

- **ServerImpl**: implementa os demais agentes RMI *servers* através da especificação do agente (classe `agent.java`), ou seja, este programa é executado diversas vezes com parâmetros diferentes, tais como, nome do agente, nome da máquina, porta;
- **BuscaMensagem**: implementa todos os tipos de mensagem mapeados no ambiente. É nesta classe que o agente reconhece e trata as mensagens, é utilizada pelos agentes do tipo `ServerImpl`;
- **MandaResposta**: é utilizada sempre que um agente do tipo `ServerImpl` deseja enviar uma mensagem.

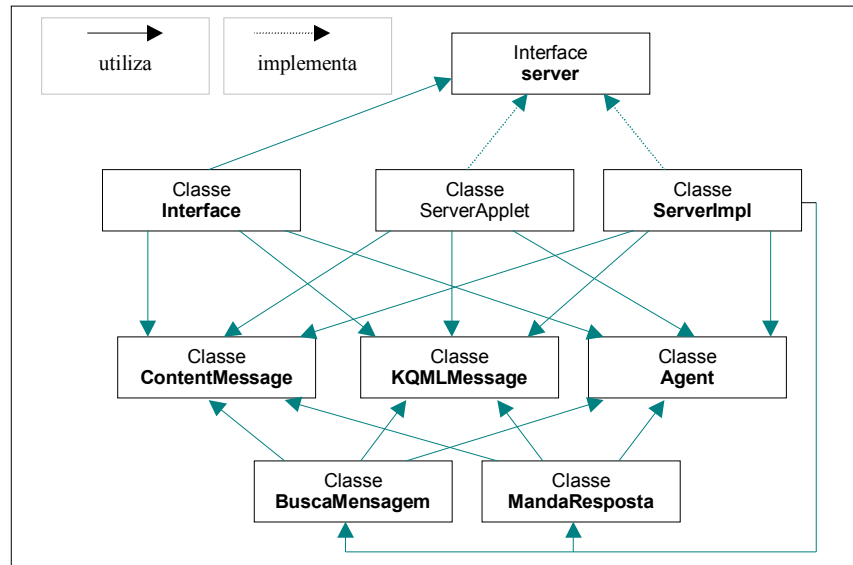


FIGURA 22 – Diagrama das Principais Classes

Além das classes descritas acima, existem outras classes complementares, como o *applet* de exercício, que consiste em um *template* que contém fórmulas para a aplicação dos exercícios numéricos. O arquivo HTML de chamada de um exercício contém o enunciado e o tipo de fórmula que deve ser utilizada, assim, o *template* gera um exercício com o enunciado especificado, valores randômicos e aplica-os à fórmula do exercício especificado para obter a resposta correta deste e podendo, assim, corrigir a resposta informada pelo aluno.

4.3 Estratégia de Ensino

O ambiente Eletrotutor foi desenvolvido de modo a poder ser utilizado pelo aluno de duas formas diferentes: o modo Tutorial e o modo Autônomo. No modo Autônomo, o aluno detém total controle sobre a sessão de estudo, podendo realizar qualquer lição, acompanhar qualquer exemplo ou fazer qualquer exercício, na seqüência que melhor lhe convier. Nesta modalidade, não é mantido qualquer registro dos dados do aluno ou das lições realizadas. A metodologia de ensino adotada é totalmente centrada no aluno.

No modo Tutorial, o sistema assume o controle da sessão, definindo a seqüência mais adequada de lições, exemplos e exercícios. Para tanto, o tutor utiliza uma

estratégia de diagnóstico através da verificação da performance deste aluno, por meio dos seus erros e acertos nos exercícios propostos.

Desta forma, o Eletrotutor adapta dinamicamente a sua estratégia de ensino, de acordo com seu próprio diagnóstico, assim como, por solicitação do aluno, explanando conceitos, gerando exemplos, propondo e corrigindo exercícios, fazendo revisões, remetendo o aluno a estudar outros pontos, enfim, procurando fazer exatamente o que um tutor humano faria para ajudar o aluno, passo a passo, a ir superando os obstáculos até atingir um patamar que ambos julguem satisfatório.

O número de acertos e erros de cada lição e o número de exemplos e exercícios realizados são utilizados como principal método de avaliação para obter o grau de compreensão do aluno. Este modelo adotado é essencialmente quantitativo, como pode ser percebido, por isso foi posteriormente aperfeiçoado no JADE, com os avanços na estrutura da modelagem do aluno.

A dinâmica do modo tutorial utilizada pelo Eletrotutor III foi implementada de acordo com [SIL 92] e funciona de um modo tal que, a cada avaliação, é proposta uma série de até, no máximo, sete vezes o mesmo tipo de exercícios, mas sempre com valores diferentes, escolhidos randomicamente em uma tabela de valores pré-determinados. O procedimento seguinte do tutor é determinado pela performance do aluno, sendo que o tutor se comporta de acordo com o resultado obtido em cada série de exercícios de um mesmo tipo, conforme um dos casos descritos na Tabela 10.

TABELA 10 – Casos de performance do Aluno

Nível	Situação	Estado do aluno
1	Três acertos consecutivos ou cinco acertos alternados em sete tentativas.	Aluno com conhecimento satisfatório.
2	Quatro acertos alternados em sete tentativas.	Aluno com conhecimento razoável.
3	Três acertos alternados em sete tentativas.	Aluno com pouco conhecimento.
4	Três erros consecutivos ou cinco erros alternados em sete tentativas.	Aluno com conhecimento insatisfatório.

Segundo a Tabela 10, no primeiro caso, o aluno encontra-se num estado que foi denominado Nível 1, no qual o tutor encaminha o aluno para a próxima série de exercícios ou para a próxima lição, dependendo do ponto onde ele se encontra. No segundo caso, denominado Nível 2, o aluno é remetido a uma nova série do mesmo tipo de exercício para submeter-se a uma nova avaliação. No terceiro, o aluno encontra-se num estado denominado Nível 3, onde o aluno é remetido novamente a uma nova série de exemplos da lição. No quarto e último caso, denominado Nível 4, ele é remetido novamente ao início da lição a fim de recapitular a matéria. Esta nomenclatura adotada é uma forma de representar o estado do aluno, durante a sua avaliação, internamente para o programa, não sendo portanto acessível ao aluno.

Deste modo, o tutor encaminha o aluno a percorrer todas as lições, executando os exercícios até que ele obtenha uma boa performance em cada um deles. No entanto, a todo momento, o aluno tem liberdade para interromper o processo tutorial, modificando

a seqüência estabelecida e acompanhando a lição que desejar, desde que a lição desejada já tenha sido previamente estudada por este aluno.

4.4 Modelo de cooperação entre os agentes

Durante a implementação do Eletrotutor, foi definida a dinâmica de funcionamento do ambiente, através de uma seqüência de troca de mensagens entre os diversos agentes utilizada no JADE. A partir daí, foi desenvolvido um modelo de Máquina de Estados para cada agente, analisando todas as situações possíveis que cada agente pode se defrontar e qual deve ser a atitude que ele deverá tomar em cada caso.

Foi também definido nesta etapa, o protocolo de troca de mensagens entre os agentes em linguagem KQML. Todas as mensagens que os agentes necessitam enviar e receber foram mapeadas, já que é através da troca das mensagens entre os agentes que o ciclo interno destes, se baseia. Portanto, os agentes sempre terão a capacidade de decompor as mensagens e decidir qual as ações pessoais e de comunicação que devem executar. Conforme já foi visto em capítulo anterior, as mensagens são formadas em duas partes: o Cabeçalho da Mensagem, de acordo com o padrão definido na especificação KQML, e o Conteúdo da Mensagem, que pode assumir variados formatos, dependendo da situação na qual o agente se encontra.

Os agentes trocam as mensagens através do *Remote Method Invocation* (RMI) [SUN 98], por permitir a passagem de objetos pelo canal de comunicação entre os agentes pois, como foi visto, as mensagens são constituídas por objetos implementados em Java.

A figura 23, mostra o fluxo de mensagens, ou seja, os passos e conteúdos de mensagens, a partir da conexão de um usuário até a recuperação do conteúdo a ser apresentado. Algumas mensagens trocadas entre os agentes (1 ao 6) são especificadas a seguir:

- (1) Agente Interface (6) envia Identificação do usuário e sua senha;
- (2) Agente Gerenciador do Modelo do Aluno (5) busca validação da identificação do usuário na base de conhecimentos. Se este usuário não existir, o Agente Gerenciador do Modelo do Aluno cadastra o novo usuário.
- Se o usuário já estiver cadastrado (2.1), são recuperadas as informações referentes ao aluno e o Agente Gerenciador do Modelo de Aluno envia a um dos agentes (1, 2 ou 3) correspondente a quem deva assumir o controle do processo, uma mensagem (3) para dar início à sessão propriamente dita. Este agente, por sua vez, recupera na sua base de conhecimentos o conteúdo que deve ser apresentado a este usuário (3.1) e o envia ao usuário (3.2). Através do Agente Interface (6) o usuário especifica o que deseja, interagindo com o sistema através da interface gráfica como, por exemplo, clicando um botão a fim de executar o próximo exercício (3.3).
- Quando a tarefa que o agente se propunha a realizar se encerra, ele avisa o agente Gerenciador do Modelo do Aluno (4), para que este com base no desempenho da última tarefa possa decidir qual o próximo passo. Antes disso, porém, é atualizado o Modelo do Aluno na base de conhecimentos.

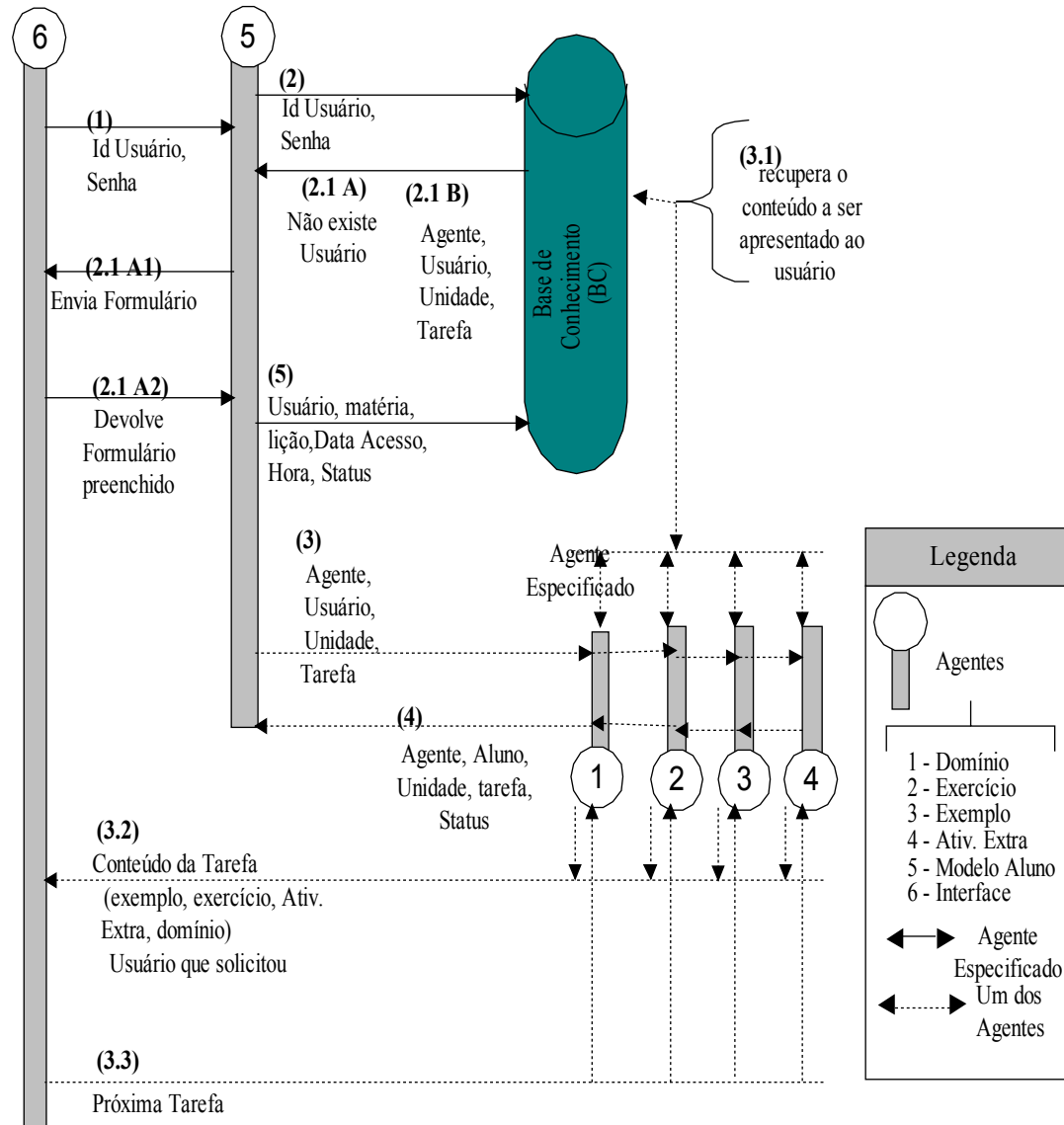


FIGURA 23 – Fluxo das Mensagens Entre os Agentes

As ações dos agentes, no Eletrotutor III, são executadas a partir das regras de comportamento gerais, tais regras definem quando uma ação deve ser executada, por exemplo, a partir de quais condições de mensagens e dos estados mentais, determinada ação é acionada. Portanto as regras de comportamento estão ligadas as mensagens KQML e aos estados mentais dos agentes.

As cinco regras básicas de comportamento são:

- **login_usuario:** esta regra ocorre quando o agente Interface envia uma mensagem ao agente Gerenciador Modelo do Aluno, informando que um aluno se conectou ao sistema. Como resultado desta regra, ocorre o cadastramento do aluno (caso o aluno não tenha sido cadastrado previamente, em uma sessão anterior), ou a recuperação do ponto do curso em que o aluno se encontrava quando encerrou a última sessão. Esta regra desencadeia a regra mostra_conteúdo;
- **mostra_conteudo:** ocorre quando o agente Gerenciador Modelo do Aluno envia uma mensagem para o agente que receberá o controle do processo que, por sua vez,

tenta encontrar o conteúdo ou a atividade a ser apresentado ao aluno, caso obtenha sucesso nesta tarefa, ele envia ao agente Interface a mensagem com a informação necessária, caso contrário, desencadeia uma mensagem ao agente Gerenciador Modelo do Aluno para que este informe qual o próximo passo a seguir;

- **próximo**: tal regra é desencadeada quando o aluno pressiona o botão próximo da interface. O agente Interface, então, envia uma mensagem ao agente que possui o controle do processo. Este, por sua vez, tenta encontrar na base de conhecimentos o próximo conteúdo ou atividade a ser mostrada, se obtém sucesso, envia tal conteúdo ao agente Interface, caso contrário, envia mensagem ao agente Gerenciador Modelo do Aluno, desencadeando assim, outra regra de comportamento: a regra “controle”;
- **anterior**: essa regra é similar à regra **próximo**, e é desencadeada quando o aluno pressiona o botão anterior da interface. O agente Interface, então, envia uma mensagem ao agente que possui o controle, este por sua vez, tenta encontrar na base de conhecimentos o conteúdo anterior ao atual, se obtém sucesso, envia tal conteúdo ao agente Interface, caso contrário, envia também a mensagem ao agente Gerenciador Modelo do Aluno desencadeando assim, a regra de comportamento: “controle”;
- **controle**: o agente Gerenciador do Modelo do Aluno recebe uma mensagem de um agente que não conseguiu recuperar o próximo conteúdo ou o conteúdo anterior ao atual. Para descobrir o conteúdo próximo ou anterior o agente deve saber qual o agente que está requisitando esta ajuda, já que, por exemplo, acabaram-se as lições o próximo conteúdo a ser mostrado, é um exemplo, se este existir, senão um exercício.

O sistema pode evoluir através da definição de novas mensagens entre os agentes bem como a inclusão de novas regras de comportamento, já que, cada regra de comportamento contém as ações que os agentes devem executar quando recebem determinada mensagem. Assim, caso sejam necessários à implantação de novos agentes ou se os agentes evoluírem nas suas atribuições, os demais agentes devem estar preparados para tratar novas mensagens que eventualmente surjam em decorrência desta evolução.

4.5 A Interface

Ao conectar-se ao sistema, utilizando um navegador (*browser*), o aluno deve acessar o endereço WWW onde encontra-se a página inicial do ambiente, inserindo a URL correspondente àquele endereço. Com isso, o aluno passa a visualizar a página HTML inicial do sistema que contém as informações que o aluno irá necessitar para dar andamento à sessão.

A interface proposta no protótipo desenvolvido por [BIC 99] é dividida em dois *frames*: um que contém o agente Interface (*Java applet*) e outro, no qual o material instrucional formado por *applets*, *JavaScript* e HTML é apresentado, como ilustram as Figuras a seguir.

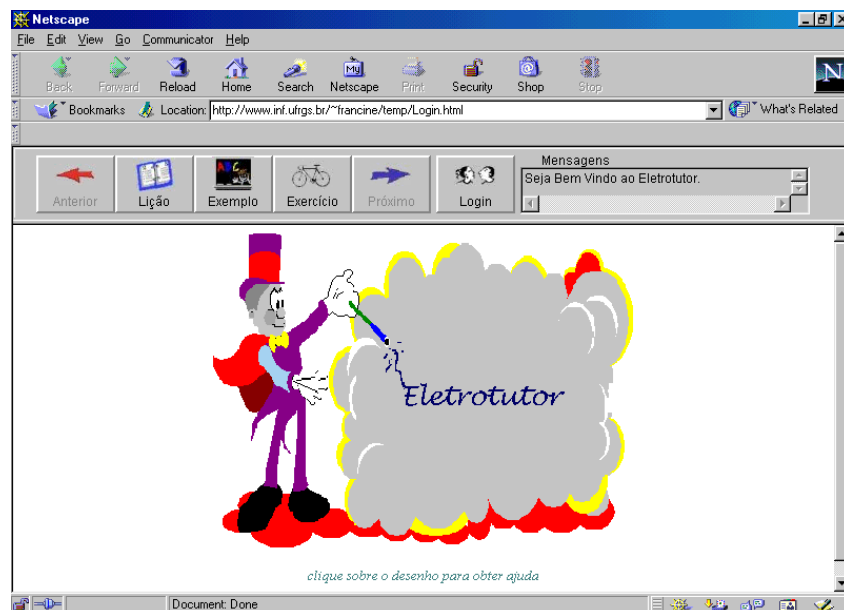


FIGURA 24 – Tela de Abertura

Esta versão possui somente o modo tutorial e os botões *Lição*, *Exemplo* e *Exercício* estão desabilitados, pois são os agentes que decidem o momento e a forma como o material instrucional é apresentado, a interface da versão posterior onde o modo autônomo foi também implementado será mostrada mais adiante.

A figura 24 representa a tela de entrada da versão do Eletrotutor apresentada por [BIC 99]. Para se conectar ao Modo Tutorial, é necessário pressionar o botão *Login*, logo a seguir, o sistema abrirá uma janela de identificação, na qual o usuário deve preencher os dados requeridos, como ilustra a figura 25. O usuário deve pressionar o botão *Ok* para confirmar seus dados e aguardar enquanto o sistema verifica a sua identificação.



FIGURA 25 - Janela de Login

Para ocorrer uma maior interação entre o aluno e os agentes foi elaborado um quadro de mensagens, no qual os agentes enviam informações ao aluno, conforme mostrado nas figuras. Tais mensagens informam ao aluno, por exemplo, qual a lição, exemplo ou exercício em que ele se encontra, assim como o andamento da correção dos exercícios.

O material instrucional apresentado é formado, conforme foi visto, por lições (páginas HTML), exemplos (páginas HTML e *JavaScript*) e exercícios (*Java applet*). A figura 26 mostra um exemplo de tela de lição.

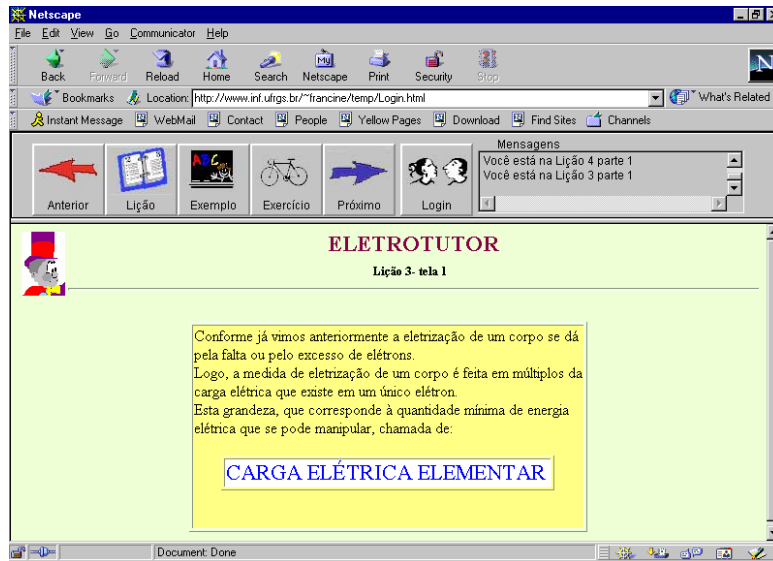


FIGURA 26 - Interface com Conteúdo de Lições

Os exemplos, conforme mostrado na figura 27 são elaborados de modo a mostrar ao aluno a aplicação de fórmulas através da solução de exercícios, nos quais o próprio aluno informa ao sistema os valores numéricos, com os quais o sistema resolve o problema. O aluno pode fazer uma infinidade de exemplos de cada tipo, introduzindo valores diferentes para que ele possa, assim, inferir os conceitos aplicados



FIGURA 27 – Interface com Conteúdo de Exemplos

Os exercícios, por sua vez, são apresentados ao aluno propondo a solução de problemas nos quais os valores propostos para cada exercícios são gerados pelo próprio sistema. A cada vez que um exercício é respondido, o sistema retorna ao usuário a resposta correta e o resultado da correção, como ilustra a figura 28. Quando o aluno

executa os exercícios, o sistema monitora o seu desempenho e, através da troca de mensagens entre os agentes, conforme foi visto, apresenta as telas de acordo com o andamento da sessão, podendo mostrar um próximo exercício, retornar aos exemplos ou à lição ou, ainda, passar à lição seguinte.

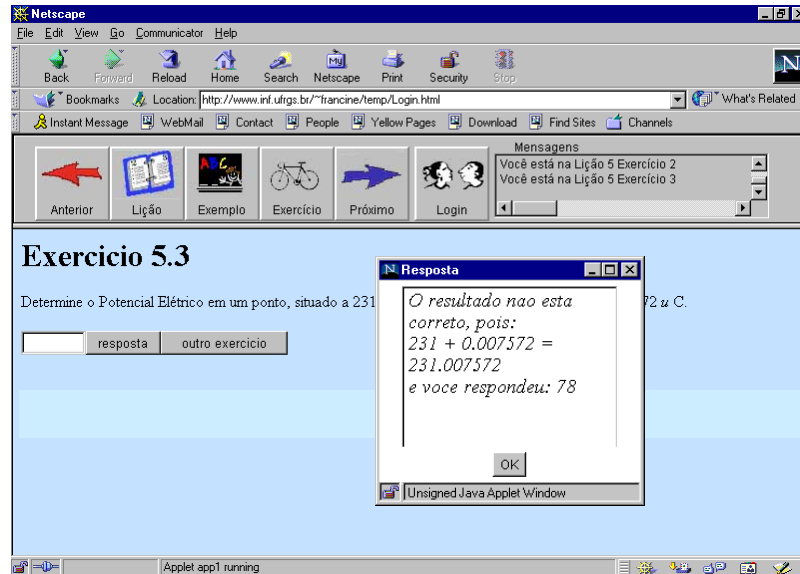


FIGURA 28 – Interface com Exercício

4.6 A evolução do Eletrotutor

Na medida em que o projeto JADE foi sendo desenvolvido, novas versões do ambiente Eletrotutor foram sendo elaboradas e, dessa forma, foi ocorrendo a evolução e o desenvolvimento do JADE.

Um trabalho completo de projeto de interface baseado em heurísticas foi apresentado por [ASC 2000]. Este trabalho teve como objetivo escrever um método heurístico que pretendesse auxiliar projetistas a desenvolver interfaces inteligentes com usabilidade e desenvolver um protótipo das interfaces do sistema Eletrotutor III, usando as técnicas de interfaces inteligentes, as heurísticas e os *guidelines* definidos no método proposto.

O Método Heurístico proposto define uma série de heurísticas que buscam facilidade de utilização, facilidade de aprendizagem, gestão e manipulação de erros, adaptabilidade. Para cada uma destas heurísticas são definidos *guidelines* que são diretrizes que orientam o projeto da interface.

A seguir, são relacionadas as heurísticas de projeto e, posteriormente, os *guidelines* propostos:

Heurísticas que buscam facilidade de utilização: uma interface deve ser:

- Eficiente
- Conveniente
- Auto-Descritiva
- Prestativa
- Confortável

Heurísticas que buscam facilidade de aprendizagem. Uma interface deve ser:

- Consistente
- Obediente
- Ter códigos e denominações significativos
- Utiliza recursos hipermídia

Heurísticas que buscam gestão e manipulação de erros. Uma interface deve ser:

- Complacente
- Tolerante a falhas do usuário
- Possuir um sistema de mensagens de erros de qualidade

Heurísticas que buscam subjetiva satisfação e atração do usuário. Uma interface deve ser:

- Prestativa e satisfazer o usuário
- Natural
- Dar *feedback* imediato
- Atender às ações explícitas do usuário
- Clara

Heurísticas que buscam adaptabilidade. Uma interface deve ser:

- Diversa
- Flexível

O trabalho apresentado propõe também um conjunto de heurísticas de avaliação para ser utilizado na metodologia de projeto de interface. São elas:

- Avaliação heurística
- Revisão de *guidelines*
- Inspeção de consistência
- Caminhos cognitivos

O método heurístico proposto por [ASC 2000] contempla um conjunto de *guidelines*, que consistem em orientações a serem seguidas pelo projetista de interface com o objetivo de atender ao conjunto de heurísticas anteriormente descrito. Estes *guidelines* classificam-se quanto:

- Facilidade aos usuários
- À adaptabilidade
- Ao *feedback*
- Às ações necessárias
- Ao controle do usuário
- À consistência
- Aos elementos da hipermídia
- Aos estilos de interação
- Às entradas de dados
- Às opções visuais disponíveis
- Aos erros e mensagens de erros
- Às formas de ajuda

O trabalho prosseguiu fazendo uma análise na versão do Eletrotutor anteriormente descrita, utilizando este método. Como resultado, um novo projeto de interface para o Eletrotutor foi obtido. A figura 29 ilustra a identificação de algumas heurísticas em uma das telas do Eletrotutor projetada por [ASC 2000]

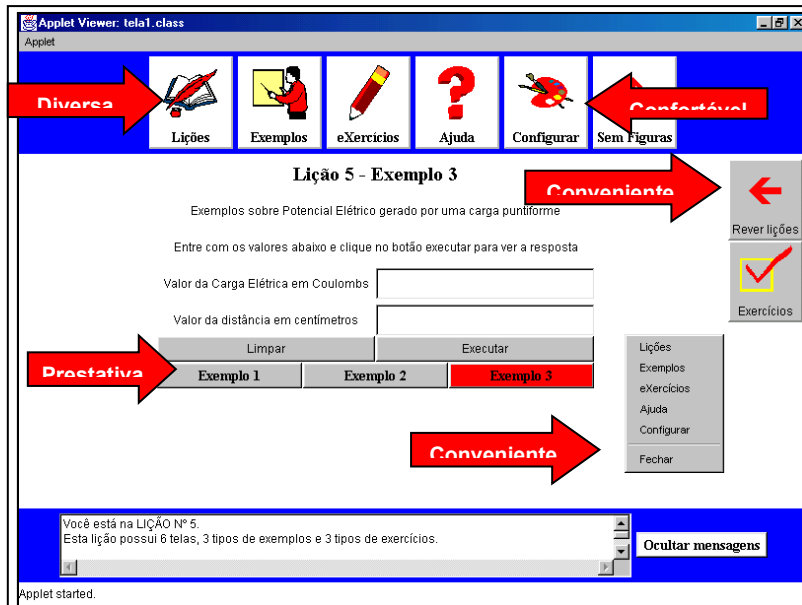


FIGURA 29 – Exemplo de Aplicação de Heurísticas

Após o projeto de uma nova interface, uma nova versão do Eletrotutor foi elaborada contemplando o novo projeto visual e incorporando as seguintes funcionalidades que foram implementadas juntamente com a evolução da arquitetura:

- implementação do modo autônomo (não-tutorial) no qual o aluno pode utilizar livremente todos os recursos didáticos disponíveis, na forma como melhor lhe convier;
- Interface dividida em três painéis: o painel superior onde são colocados os botões da barra de ferramentas principal, o painel direito onde são colocados os botões de acordo com as circunstâncias, um painel inferior em que fica a caixa de mensagens ao usuário e o painel central onde são mostradas as telas propriamente ditas dos recursos do tutor;
- Inclusão de botões de navegação no painel central junto com as telas de lições exemplos e exercícios que, além de dar maior flexibilidade ao aluno, no modo autônomo, permitindo que ele navegue livremente e de modo não seqüencial, de uma tela para outra dentro de cada lição, situa o usuário mostrando quantas telas de cada atividade existem em cada unidade.
- recursos de configuração da interface no que diz respeito a cores de

As figuras 30, 31 e 32, a seguir, mostram respectivamente as telas de abertura, de lições e de exercícios.

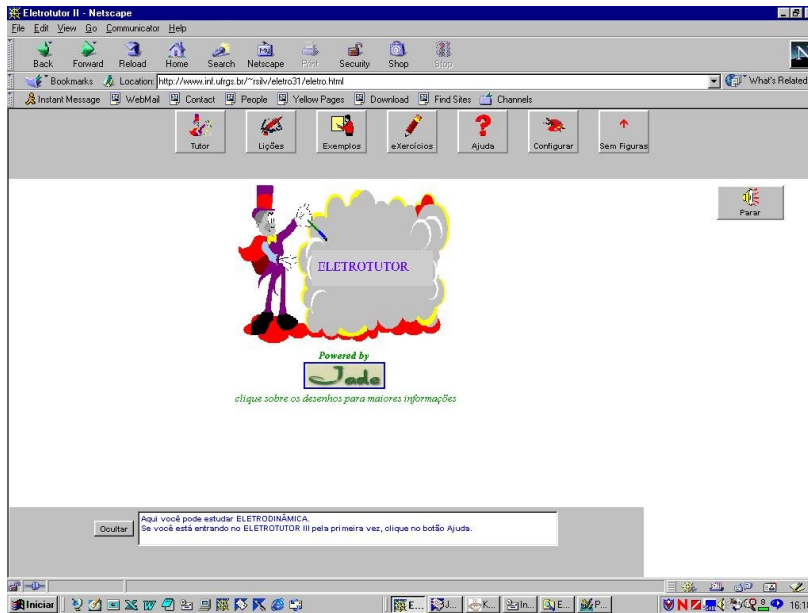


FIGURA 30 – Tela de Abertura do Elettrotutor III

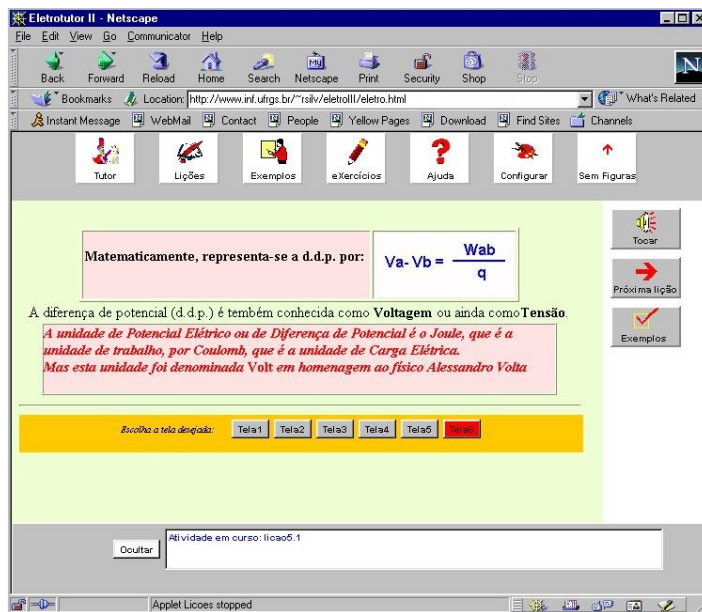


FIGURA 31 – Tela de Lição do Elettrotutor III

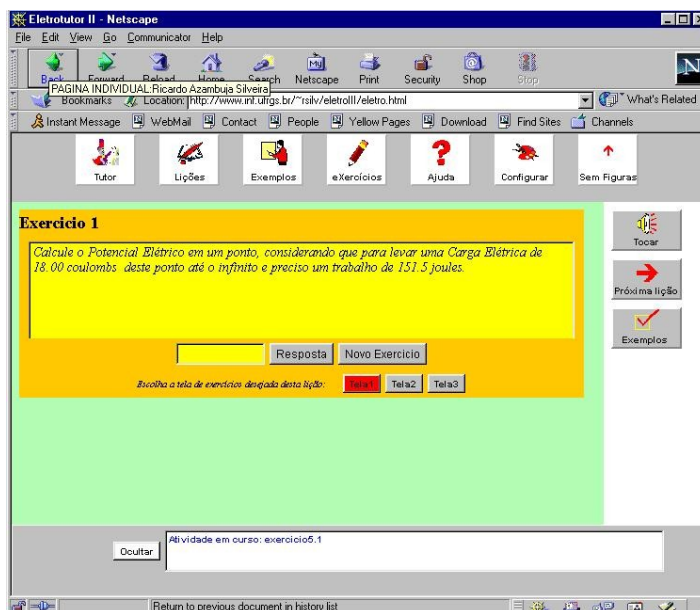


FIGURA 32 – Tela de Exercício do Elettrotutor III

Após terem sido feitas estas melhorias na versão três do Elettrotutor, uma Quarta versão passou a ser implementada, durante o que as últimas definições na implementação da arquitetura do JADE foram sendo incorporadas. Esta evolução, apresentada em [REI 2000] será resumida a seguir.

4.7 O Elettrotutor IV

A proposta de [REI 2000] é, além de contribuir na evolução da arquitetura do JADE, aperfeiçoar os mecanismos de modelagem do aluno e das estratégias de ensino, tornando o sistema capaz de implementar ambientes de ensino mais adaptativos.

As principais contribuições desta versão foram:

- Solidificação do conceito de uso de somente três tipos arquitetura de agentes: o Agente Pedagógico, o Gerenciador do Modelo de Aluno e o Gerenciador de Comunicação e o Gerenciador de Interface do Aluno;
- Melhor encapsulamento das funcionalidades dos agentes e melhor segmentação dos códigos através da implementação das classes de acordo com os papéis de cada tipo de agente;
- Nova estrutura de dados contemplando a nova estrutura interna dos agentes e as novas funcionalidades do sistema como um todo;
- Novas funcionalidades que aumentam as possibilidades de implementação de táticas de ensino, tais como uso de ferramenta própria de *chat*, botão de solicitação de revisão de conteúdo (“não entendi”), possibilidade de conectar com outras páginas externas ao ambiente;
- Implementação da Ferramenta Administrativa para a elaboração da estrutura dos cursos.

Serão descritas, na seqüência, as principais contribuições feitas durante a implementação deste protótipo.

4.7.1 A arquitetura do Eletrotutor IV

A sociedade de agentes proposta por [REI 2000] contempla os agentes definidos na arquitetura do JADE de acordo com a figura 33.

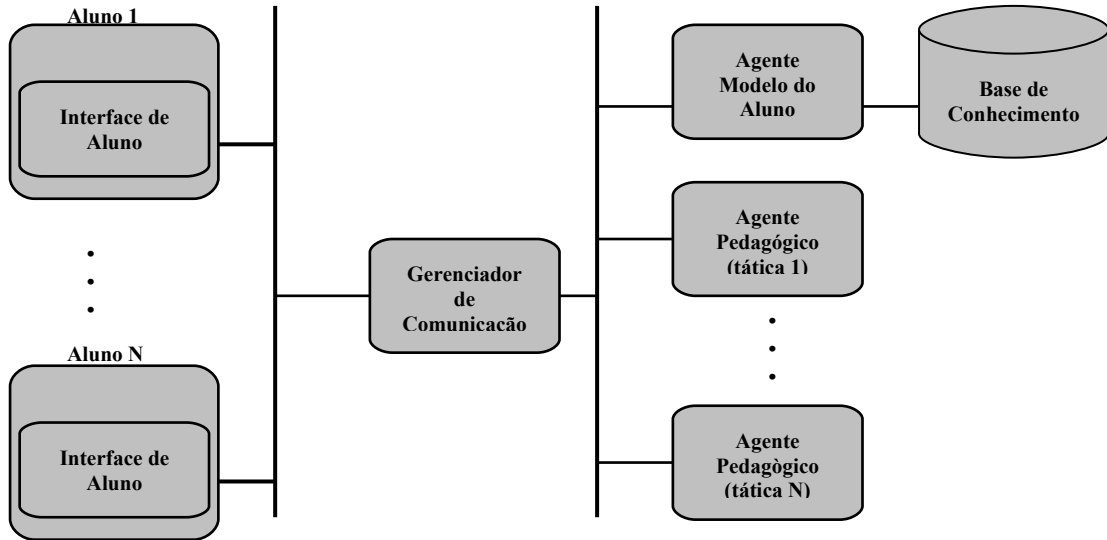


FIGURA 33 - Arquitetura do Sistema Tutorial

O Eletrotutor IV manteve assim a arquitetura com um agente responsável pelo controle geral do sistema de multiagentes, denominado Gerenciador do Modelo do Aluno. A comunicação entre os agentes é também coordenada por um agente Gerenciador de Comunicação. Os demais agentes, denominados Agentes Pedagógicos são responsáveis pelas tarefas relacionadas com cada uma das táticas de ensino, na qual cada um pode ter suas tarefas específicas de acordo com o seu propósito. Estes agentes, possuem habilidades próprias inerentes às tarefas relacionadas às táticas que ele implementa. Sendo assim, funções como mostrar o conteúdo atual, avançar no conteúdo, mostrar conteúdo anterior, e outras, já são incorporados às tarefas do agente.

O agente Interface do Aluno possui como função o controle do *browser* no ambiente do aluno. As mensagens são recebidas e enviadas a todos os agentes através do agente Gerenciador de Comunicação, ou seja, o agente apenas repassa as atividades do aluno na interface do sistema e recupera as informações a serem retransmitidas aos alunos.

O agente Gerenciador de comunicação, por sua vez, possui a função de controlar a comunicação com os demais agentes e encaminhá-los para alguma ação acionada pelo aluno no tutor. As mensagens são recebidas e enviadas a todos os agentes através de Java/RMI.

O Agente Modelo do Aluno é responsável por construir e manter uma base de conhecimento que represente o estado cognitivo dos alunos que estejam conectados ou que tenham estado conectados ao sistema. Quando a sessão tutorial inicia, este agente

recebe informações sobre a identificação do aluno e recupera as informações correspondentes a ele na sua base de conhecimento.

Primeiramente, o Agente Modelo do Aluno procura validar o *login* do aluno em um determinado curso. Se, ao recuperar as informações sobre o aluno, o agente constatar que nenhum agente pedagógico havia sido iniciado em uma sessão anterior, é, então, apresentado um conjunto de opções de táticas ao aluno para que este possa escolher como deverá ser o início de seu estudo. Após esta primeira fase, o agente selecionado passa a ser o responsável por todas as novas requisições efetuadas.

Todas as funções de acesso aos dados do aluno foram, nesta versão, concentradas no Agente Modelo do Aluno. Assim, sempre que houver a necessidade de um Agente Pedagógico atualizar o histórico do aluno, este agente repassa as informações ao Agente Modelo do Aluno para que este possa atualizar a base de conhecimentos. Funções, tais como: verificar as táticas para uma determinada lição, verificar último acesso ou emitir relatório do aluno, também são executadas por este agente.

Os Agentes Pedagógicos são gerados a partir de uma tática previamente definida pelo especialista do curso. Suas tarefas serão definidas conforme a necessidade do agente. Porém, como o tutor baseia-se na apresentação dos conteúdos através da apresentação de páginas HTML, algumas tarefas já são previamente definidas para todos os agentes genéricos, entre as quais destacam-se:

- **Mostrar conteúdo atual:** significa que o agente deve recuperar na base de conhecimento qual o conteúdo atual que está sendo apresentado ao aluno e repassá-lo ao Gerenciador do Modelo de Aluno para este encaminhar ao aluno.
- **Avançar:** significa que o agente deve recuperar na base de conhecimento qual o próximo conteúdo a partir do conteúdo atual que está sendo apresentado ao aluno e repassá-lo ao Gerenciador do Modelo de Aluno para este encaminhar ao aluno.
- **Voltar:** significa que o agente deve recuperar na base de conhecimento qual o conteúdo anterior a partir do conteúdo atual que está sendo apresentado ao aluno e repassá-lo ao Gerenciador do Modelo de Aluno para este encaminhar ao aluno.
- **Opções:** significa que o agente pode ter relação com opções a serem apresentadas ao aluno, porém não é obrigatório que este agente contenha opções, mas a tarefa é implementada para recuperar quais as opções que este agente pode oferecer ao aluno, bem como invocar a ação que uma opção pode proporcionar.
- **Atualizar histórico:** a cada tarefa implementada, deve se registrar no Modelo de Aluno as ações que este está realizando sobre o tutor. Por exemplo, no caso de mostrar o próximo conteúdo, o agente seta o histórico do aluno com a data e hora de saída de conteúdo atual e seta o histórico do aluno com a data e hora de entrada de um próximo conteúdo.
- **Comunicação:** o agente implementa uma função que localiza onde o Gerenciador de Comunicação se encontra, para repassar informações.

- **Avaliação:** significa que o agente deve ter um mecanismo de avaliar o aluno para a tática que o agente representa. Esta tarefa de avaliar não segue um padrão para todos os agentes, mas deve ter a obrigatoriedade de sua implementação.

Toda a estrutura de comunicação utilizada na versão anterior foi mantida. Entretanto, devido às alterações na estrutura da base de dados desta versão, foi necessário modificar a estrutura da classe *ContentMessage*, anteriormente descrita, a fim de contemplar esta nova estrutura. A classe *Content* utilizada no Eletrotutor IV é mostrada na figura 34.

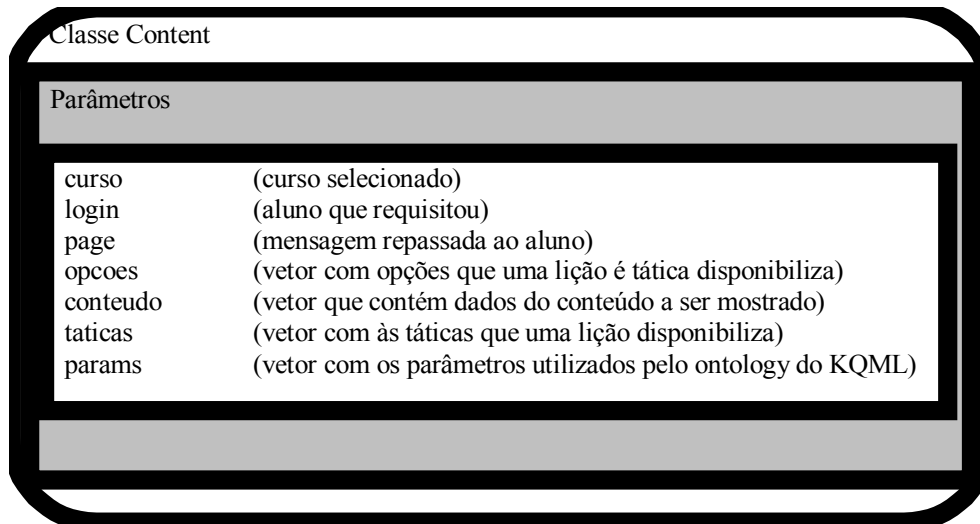


FIGURA 34 – Classe *Content*

4.7.2 As Estratégias de ensino e o Modelo do Aluno

As estratégias de ensino no Eletrotutor IV são executadas de acordo com o repertório de táticas desenvolvidas pelo especialista, durante a elaboração do curso, através da inserção das informações na base de dados relativas ao Modelo de Estratégias, utilizando a Ferramenta Administrativa.

Conforme a ação do estudante durante a sessão de ensino, o tutor pode mudar de tática alterando a forma de apresentação de uma determinada lição ou apresentando algum outro tipo de atividade diferente daquela que vinha sendo feita, desde que o autor do curso tenha disponibilizado diversas táticas alternativas referentes àquela lição. Esta ação é executada a partir de regras de comportamento inseridas no Modelo de Estratégias pelo autor do curso, através das quais o Gerenciador do Modelo de Aluno recorre aos agentes pedagógicos solicitando o início de uma nova atividade para aquele aluno.

Estas ações são executadas pelo sistema baseado nas avaliações constantes que são feitas, e que são inerentes a cada tática de ensino. As avaliações servem para determinar o que o aluno tem aprendido sobre o domínio. A proposta do Eletrotutor IV, é que para cada tática desenvolvida, o especialista determine um método de avaliação adequado. Para efetuar tais avaliações é necessário que o especialista determine um valor quantitativo para cada lição e tática apresentadas. Este valor é determinado como PESO de avaliação.

Na atual versão do Eletrotutor, existem duas formas de obter este indicador:

- **Regras do Histórico:** o sistema possui duas regras específicas sobre o histórico do aluno:
 - **pelo número de acesso** do aluno em determinado conteúdo.
 - **pelo tempo utilizado** pelo aluno para compreender este conteúdo.

Assim, o especialista pode montar regras e determinar o seu peso. Por exemplo: se o aluno acessou tais conteúdos e, para cada conteúdo, levou determinado período de tempo, o aluno obtém determinado PESO, pois o especialista pode prever que para uma determinada lição é obrigatório o aluno passar por determinadas páginas, assim como para uma determinada página seria impossível o aluno aprender em pouco espaço de tempo.

- **Táticas de avaliação:** este método avalia o aluno através de alguma forma objetiva de quantificar respostas informadas por ele nos exercícios propostos.

Além de determinar algum método de avaliação, o especialista deverá também estabelecer regras que determinem as ações que o sistema deve tomar em determinada lição, de acordo com cada tática e o resultado da avaliação obtido pelo aluno.

Outro recurso disponibilizado no Eletrotutor IV foi a disponibilidade ao aluno de recursos e ferramentas.

O especialista pode determinar, para um determinado conteúdo a ser apresentado, que recursos irá disponibilizar ao aluno. Estes recursos serão apresentados na forma de link na interface, na qual terá como objetivo três tipos de ações:

- **Mudança de estratégia:** o especialista pode criar recursos como por exemplo, a do aluno não estar entendendo a forma de como está sendo apresentada determinada lição. Por exemplo, pode-se criar um link “não entendi” e o tutor ao verificar esta chamado por parte do aluno, acionar uma estratégia relacionada a esta ação e mudar a maneira de apresentar esta lição.
- **Ferramentas locais:** estas ferramentas apesar de serem apresentadas diretamente no *browser*, não terão o conceito de rede, em virtude de não haver comunicação externa com outros pontos. Estas ferramentas são:
 - **Ajuda:** explica ao aluno a funcionalidade da interface, do tutor, etc.
 - **Dicas:** dicas para resolver um determinado exercício, aprender mais sobre um determinado conteúdo, etc.
 - **Calculadora:** utilizada para a resolução de exercícios.
- **Ferramentas na rede:** o Eletrotutor IV disponibiliza ao aluno determinadas ferramentas de rede para que ele possa ter mais recursos para tirar suas dúvidas ou obter conhecimento adicionais sobre determinado tópico.. Estas ferramentas são Sala de reuniões virtuais (*chat*), Fórum, Busca e Links Auxiliares

A figura 35 mostra a interface da ferramenta administrativa, implementada junto com o Eletrotutor IV

Its na Web - Ferramenta Administrativa - Microsoft Internet Explorer

Arquivo Editar Exibir Favoritos Ferramentas Ajuda

Voltar Pesquisar Favoritos Histórico

Endereço C:\download\frameprincipal.html

Ferramenta Administrativa

CURSOS	inserir	consultar	ALUNOS	inserir	consultar
LIÇÕES	inserir	consultar	TÁTICAS	inserir	consultar
OPÇÕES	inserir	consultar	ESTRATÉGIAS	inserir	consultar
CURSOS-ALUNOS	inserir	consultar	LIÇÕES-TÁTICAS (Conteúdo)	inserir	consultar
LIÇÕES-TÁTICAS-OPÇÕES	inserir	consultar			

Inserir Lições-Táticas (Conteúdo): representa o conteúdo que deve ser apresentado ao aluno conforme a lição e tática utilizada. Este conteúdo é apresentado em uma página html.

Código da Lição (Informe o código da lição):	<input type="text" value="1"/>
Código da Tática (Informe o código da tática associada a esta lição):	<input type="text" value="1"/>
Ordem Página (Informe qual a seqüência deste conteúdo):	<input type="text" value="1"/>
Protocolo (Informe "o:" se o conteúdo é carregado local ou "http://" se for público na Web):	<input type="text" value="http://"/>
Endereço (Informe o endereço que a página estará localizada com a barra no final. Ex.: /Eletrotutor/Conteudo/):	<input type="text" value="/Eletrotutor/Conteudo/"/>
Página (Informe o nome da página para este conteúdo):	<input type="text" value="conteudo111.html"/>
Avaliador (Informe "S" se o conteúdo é de avaliação ou "N" caso contrário):	<input type="text" value="N"/>
Peso (Informe "0" se o avaliador é "N" ou o peso do conteúdo):	<input type="text" value="0"/>

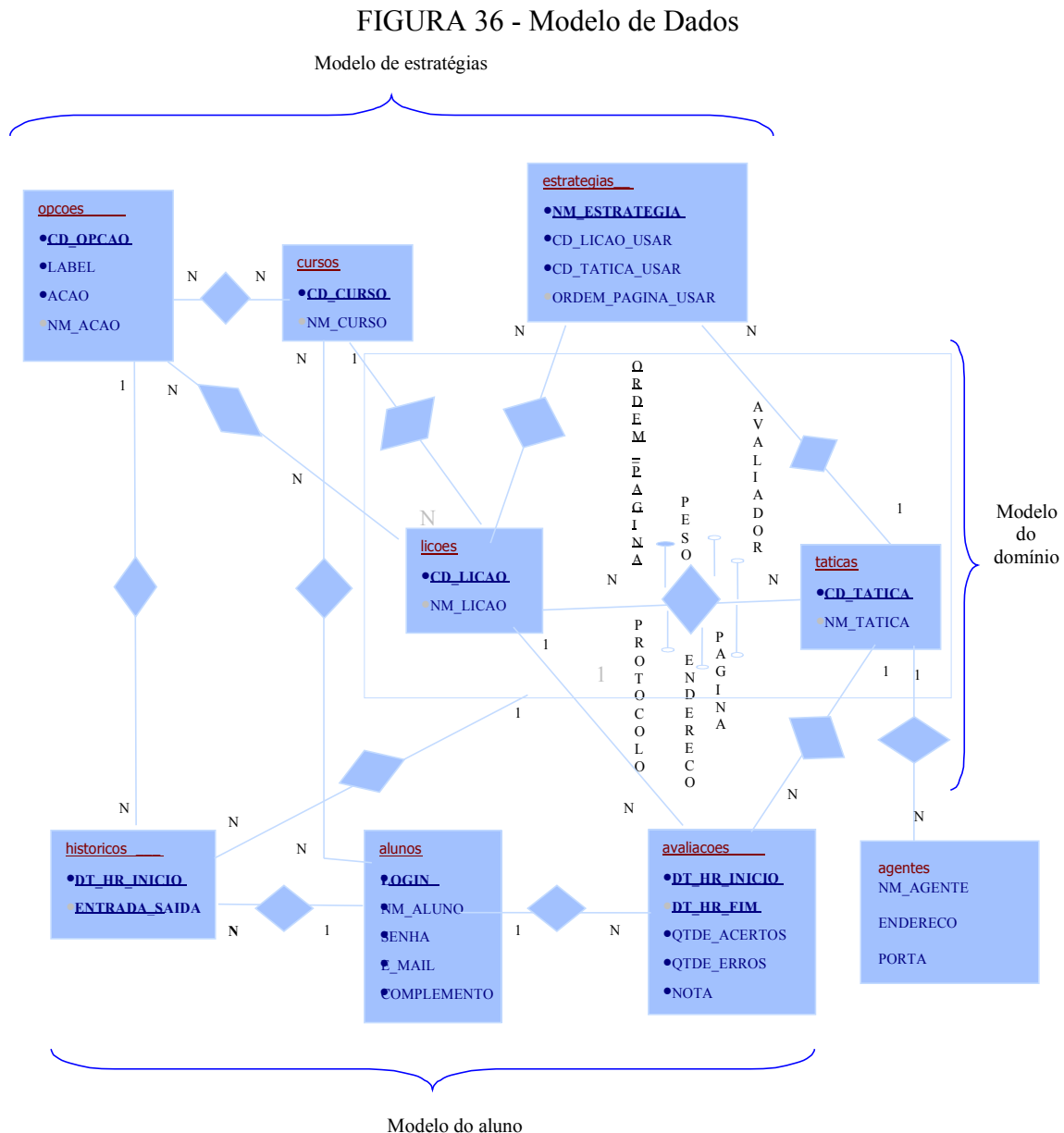
(Clique aqui para inserir o conteúdo)

Meu computador

FIGURA 35 – Ferramenta Administrativa do JADE.

4.7.3 Modelo Entidade-Relacionamento (ER)

A figura 36 mostra o modelo relacional da base de dados empregada pelos agentes do JADE



Para armazenar todas as informações necessárias para a execução do sistema, desde à apresentação do conteúdo e o armazenamento dos dados do aluno, foi necessário criar um modelo de Entidade e Relacionamento (ER) conforme mostrado na figura 32. Este modelo é composto por tabelas ditas básicas e compostas descritas respectivamente nos próximos itens. Este modelo comporta vários cursos, portanto, poderá ser utilizado para várias propostas/implementações de ambientes de ensino/aprendizado que comportem esta modelagem.

- Tabela “cursos”: representa os cursos que podem ser utilizados no ITS proposto.

- CD_CURSO: Código do curso (chave primária)
- NM_CURSO: Nome do curso
- Tabela “alunos”: é utilizada para cadastrar o aluno, para que este possa utilizar o modo tutorial do sistema.
 - LOGIN: Identificador do aluno (chave primária)
 - NM_ALUNO: Nome do aluno
 - SENHA: Senha do aluno
 - E_MAIL: E-mail do aluno para contato
 - COMPLEMENTO: Informações complementares como endereço, formação, conhecimentos, etc.
- Tabela “cursos_alunos”: representam quais os alunos que fazem parte de um determinado curso.
 - #CD_CURSO: Código do curso (chave primária e estrangeira)
 - #LOGIN: Identificador do aluno (chave primária e estrangeira)
- Tabela “liceos”: representa qual as lições que fazem parte de um determinado curso.
 - #CD_CURSO: Código do curso (chave primária e estrangeira)
 - CD_LICAO: Código da lição (chave primária)
 - NM_LICAO: Nome da lição
- Tabela “taticas”: representa as diferentes formas de se apresentar uma lição.
 - CD_TATICA: Código da tática (chave primária)
 - NM_TATICA: Nome da tática
- Tabela “liceos_taticas”: representa as diferentes táticas de ensino utilizadas por cada lição e o conteúdo que deve ser apresentado ao aluno. Este conteúdo é apresentado em uma página HTML, criada pelo especialista. O conteúdo ainda pode representar uma página de avaliação, pois cada tática utilizada em uma lição, deve apresentar uma sistemática de avaliar se o aluno aprendeu a lição utilizando esta tática. Por exemplo, para avaliar a tática de domínio o especialista pode criar uma página com perguntas e avaliar as suas respostas.
 - #CD_CURSO: Código do curso (chave primária e estrangeira)
 - #CD_LICAO: Código da lição (chave primária e estrangeira)
 - #CD_TATICA: Código da tática (chave primária e estrangeira)
 - ORDEM_PAGINA: Indica a ordem da página para determinada lição e tática. Esta ordem pode ser utilizada, por exemplo, para indicar o próximo conteúdo a ser apresentado quando o aluno acionar a opção de avançar no aprendizado. (chave primária)
 - PROTOCOLO: Conjunto de três (3) opções:
 - *http://* - indica acesso direto a uma URL
 - *https://* - indica acesso direto a uma URL por site seguro

- *c: /* - indica acesso local a disco, ou seja, por questão de segurança o especialista não deseja que o aluno acesse o conteúdo por URL (caso este descubra o seu caminho) e apenas o sistema consiga carregar a página (utilizando a tecnologia de *servlets*).
- ENDERECO: Local onde a página estará localizada no servidor *Web* ou no disco local.
- PAGINA: Nome da página que contém o conteúdo a ser apresentado de acordo com uma lição e tática.
- AVALIADOR: Conjunto de duas (2) opções:
 - S – indica que a página apresentada é de avaliação;
 - N – indica que a página apresentada não é de avaliação.
- PESO: Indica o peso da avaliação (caso a página apresentada seja uma página de avaliação). Varia de 0 a 10.
- Tabela de “opcoes”: permite que para cada lição associada a uma tática, possa-se aumentar as opções de tela para o aluno. Estas opções representam links para determinados recursos e/ou ferramentas locais ou na Internet. Esta tabela é de extrema importância, pois o especialista pode determinar novos rumos de ensino. Por exemplo, o especialista pode criar a opção “não entendi” ou “está difícil” que faz com que o ITS utilize uma nova estratégia de ensino caso esta opção seja acionada pelo aluno indicando que este não está compreendendo a forma de como está sendo apresentado um determinado conteúdo. Outras alternativas para o especialista, é a criação de links que acionam ferramentas na rede. O atual ITS suporta algumas como: *chats*, fórum de discussão, troca de mensagens, entre outras (ver capítulo XX). Porém o sistema fica aberto a chamada de outras ferramentas na rede inclusive registrando no histórico do aluno.
 - #CD_OPCAO: Código da opção (chave primária)
 - LABEL: Descrição da opção que fica visível para o aluno
 - ACAO: Nome da lição
 - NM_ACAO: Conjunto de duas (2) opções:
 - ESTRATEGIA – indica que uma nova estratégia deve ser acionada;
 - URL – indica a chamada de um recurso na Internet.
- Tabela de “licoes_taticas_opcoes”: representa as opções que podem fazer parte de um determinado conteúdo.
 - #CD_CURSO: Código do curso (chave primária e estrangeira)
 - #CD_LICAO: Código da lição (chave primária e estrangeira)
 - #CD_TATICA: Código da tática (chave primária e estrangeira)
 - #ORDEM_PAGINA: Indica qual a página irá receber determinada opção (chave primária e estrangeira)
 - #CD_OPCAO: Código da opção (chave primária e estrangeira)

- Tabela de “estratégias”: é o ponto chave para determinar as ações que o tutor irá tomar (como ensinar o curso) para cada aluno individualizado. As estratégias também podem ser vistas como regras. A tabela verifica qual a tática e lição atualmente utilizada pelo aluno (através da tabela de históricos) e conforme for o caso, altera para uma nova lição, para uma nova tática ou para um outro conteúdo. Existem duas (3) estratégias conhecidas internamente no tutor adotadas para qualquer curso:
 1. Início: inicialmente o tutor apresenta quais as táticas existentes para apresentar a primeira lição a fim do aluno selecionar uma (a que mais lhe agrada) iniciando o modo tutorial do sistema.
 2. Avançar/Próximo: apresenta o próximo conteúdo conforme a ordem da lição e conteúdo.
 3. Retroceder/Anterior: apresenta o conteúdo anterior conforme a ordem da lição e conteúdo.
- #CD_CURSO: Código do curso (chave primária e estrangeira)
- #CD_LICAO: Código da lição (chave primária e estrangeira)
- #CD_TATICA: Código da tática (chave primária e estrangeira)
- NM ESTRATEGIA: Nome da estratégia. Este campo é importante pois relata algum acontecimento ocorrido no ensino, como “não entendi”, “está difícil”, “errou avaliação”, etc. (chave primária)
- CD_LICAO_USAR: Indica qual a lição a ser apresentada conforme a estratégia a ser adotada. Pode ser a mesma lição ou inclusive alterar para uma próxima.
- CD_TATICA_USAR: Indica qual a tática a ser adotada conforme a estratégia a ser adotada. Pode ser a tática atual como pode haver alteração na forma de apresentar determinado conteúdo.
- ORDEM_PAGINA_USAR: Índice da página para qual o CD_LICAO_USAR e CD_TATICA_USAR irão utilizar. Conjunto de cinco (5) opções:
 - Primeira – mostra a ordem 1;
 - Última – mostra a n ordem;
 - +1 – mostra a próxima página;
 - -1 – mostra a página anterior;
 - Índice qualquer – a ordem escolhida pelo especialista.
- Tabela de “avaliacoes”: representa o resultado da avaliação do aluno, em determinado tempo, tática e lição. Esta tabela pode ser vista como histórico do aluno, porém para evitar desperdício de campos, foi criada esta nova tabela (a maioria do conteúdo acessado pelo estudante não será uma avaliação).
 - #CD_CURSO: Código do curso (chave primária e estrangeira).
 - #CD_LICAO: Código da lição (chave primária e estrangeira).
 - #CD_TATICA: Código da tática (chave primária e estrangeira).
 - #LOGIN: Identificador do aluno (chave primária e estrangeira).

- DT_HR_ACESSO: Em que momento de tempo o aluno começou ou terminou de efetuar a avaliação (chave primária).
 - ENTRADA_SAIDA: Indica se a data a ser registra é de entrada (E) ou final (S) da avaliação.
 - QTDE_ACERTOS: Quantidade de acertos.
 - QTDE_ERROS: Quantidade de erros.
 - NOTA: Indica a nota final para o aluno
 - Tabela de “historicos”: representa todo o acesso efetuado pelo aluno em relação ao conteúdo do curso e as opções acessadas.
 - #CD_CURSO: Código do curso (chave primária e estrangeira).
 - #CD_LICAO: Código da lição (chave primária e estrangeira).
 - #CD_TATICA: Código da tática (chave primária e estrangeira).
 - #ORDEM_PAGINA: Indica qual a página que o aluno acessou em determinada lição e tática. (chave primária e estrangeira).
 - #CD_OPCAO: Código da opção (chave primária e estrangeira).
 - #LOGIN: I dentificador do aluno (chave primária e estrangeira).
 - DT_HR_ACESSO: Em que momento de tempo o aluno começou a efetuar a avaliação (chave primária).
- ENTRADA_SAIDA: Indica se a data a ser registra é de entrada (E) ou final (S) da avaliação.

5 Considerações finais

Ao encerrar este trabalho, cabe rever as motivações que nos levaram a empreendê-lo, se os objetivos buscados foram alcançados, que contribuições avalia-se que este trabalho possa ter trazido e que trabalhos futuros podem advir dele.

Quando a primeira versão do Eletrotutor foi idealizada em 1992, durante o curso de Mestrado em Educação, tomou-se contato pela primeira vez com a idéia da utilização de conceitos e técnicas de Inteligência Artificial no desenvolvimento de sistemas de ensino auxiliados por computador. Nosso interesse por este assunto, entretanto, originou-se em meados de 1989, através de um programa pioneiro, para a época, de capacitação e desenvolvimento de recursos humanos através de cursos produzidos para treinamento na área de sistemas de telecomunicações.

Daquela época até hoje a tecnologia evoluiu no que diz respeito ao hardware, ao desenvolvimento das redes de computadores, e à popularização da Internet, como o próprio uso do computador, propriamente dito, tornou-se bastante familiar.

No entanto, se for comparada a evolução dos sistemas de ensino por computador com a evolução da Ciência da Computação como um todo, observa-se que pouca coisa mudou e muito ainda há para ser feito. Evidentemente, que a Informática na Educação beneficiou-se diretamente com a evolução no campo da Multimídia, das Redes e da agilidade e do acesso à informação. No entanto, pouca diferença estrutural existe entre a grande maioria do software educacional disponível no mercado, além da incorporação das tecnologias citadas.

Tudo isso nos motivou a desenvolver um trabalho que trouxesse uma contribuição, mínima que fosse, no sentido de mudar este contexto. Várias alternativas foram inicialmente levadas em conta: pensou-se em desenvolver um trabalho sobre modelagem cognitiva de aluno, aproveitando a experiência adquirida no programa de mestrado, pensou-se em propor avanços específicos na área de sistemas multiagentes, desenvolvendo protocolos de cooperação entre agentes pedagógicos, entre outras.

Todas as alternativas de trabalho imaginadas, no entanto, pressupunham a existência de uma arquitetura geral de um ambiente de ensino inteligente que estivesse pronta e objetivavam avançar e sobre algum componente desta arquitetura, aprofundando, no campo da pesquisa teórica, algum tema que contribuísse para a solução de um problema prático que pudesse trazer avanços na arquitetura como um todo.

Entretanto, encontramos dificuldades para encontrar uma arquitetura onde se pudesse desenvolver um trabalho pontual mais profundo. Conforme foi discutido na introdução, a maioria dos projetos de desenvolvimento de ambientes de ensino permanecerem restritos aos ambientes de laboratório, devido ao fato de que várias tecnologias tem sido desenvolvidas sem, no entanto, serem agregadas em um projeto único que aproveite o melhor do estado da arte de cada uma delas e que possa evoluir à medida que cada uma delas evolui.

As pesquisas sobre ambientes inteligentes e distribuídos de ensino são, de um modo geral, desenvolvidos como aplicações customizadas, ou seja, projetos *ad-hoc*, feitos sob medida para os fins a que se propõem e raramente são desenhados para evoluir, como um processo de refinamento iterativo;

À medida que isto vinha sendo constatado, nasceu então a idéia de desenvolver o embrião de um *framework* que, antes de evoluir para ser uma ferramenta de desenvolvimento de software educacional, seja, acima de tudo, um ambiente e uma idéia, em torno da qual diversos projetos de pesquisa pura e aplicada voltadas para o desenvolvimento de tecnologias para a implementação de sistemas distribuídos e inteligentes de ensino possam ser realizadas.

Nasceu então o JADE, que parece que vem cumprindo este objetivo, pois em conjunto com este trabalho, um outro de conclusão de graduação em Computação, uma monografia de curso de especialização em Psicologia e três dissertações de mestrado já foram produzidas, levando a crer que o principal objetivo foi alcançado.

O objetivo deste trabalho, formalmente, foi propor uma infra-estrutura de projeto, desenvolvimento e implementação de sistemas de Ambientes Inteligentes Distribuídos de Aprendizagem baseado em abordagem de Arquiteturas Multiagentes e voltado essencialmente para a Educação a Distância. Para tanto, foram estabelecidos como objetivos específicos, propor uma arquitetura de uma sociedade de agentes, que possa ser aplicada para o projeto de ambientes de ensino em diversos domínios, uma arquitetura interna de agente, propor a utilização de mecanismos de comunicação e de cooperação entre agentes adequados a uma sociedade de agentes pedagógicos, propor avanços na modelagem de estados mentais internos dos agentes e trazer contribuições de técnicas de ensino apropriadas.

Ao fazer uma avaliação da consecução de cada um destes objetivos, percebe-se que ainda há bastante a ser feito, considerando cada um deles individualmente, apesar de alguns avanços importantes. Se considerarmos o conjunto deles, entretanto, avalia-se que foi lançada uma semente sobre cada um deles e, se tivermos dificuldades para idealizar um trabalho específico, focando em aprofundar uma teoria ou uma técnica específica, acredita-se que o JADE possa, daqui para frente, desencadear ainda uma série de trabalhos em cada uma dessas áreas, e considera-se essa maior contribuição desta tese.

Cabe ainda registrar algumas contribuições específicas e avanços significativos realizados neste trabalho:

- o estabelecimento de uma estrutura nova básica (*framework*) de agentes para o desenvolvimento de ambientes inteligentes de ensino e que possa ser efetivamente empregada para o desenvolvimento de ambientes voltados para múltiplos domínios;
- o estabelecimento de uma metodologia funcional de projeto de ambientes inteligentes, desde a modelagem até a implementação do ambiente, com base nesta estrutura;
- a implementação de ambientes de ensino inteligentes com uma sociedade de agentes em uma arquitetura na qual estes agentes sejam realmente autônomos e independentes e que estejam fisicamente distribuídos em diferentes servidores, isto é, suportada por uma estrutura de rede e tirando real proveito dela;
- a proposição mecanismos de coordenação da ação dos agentes adequados à ação de agentes pedagógicos e a implementação de uma estrutura formal de comunicação que suporte estes mecanismos;
- a implementação de um ambiente real em I.A.D. na área de Tutores Inteligentes propiciando uma avaliação da metodologia e da estrutura proposta;

- a proposição de uma arquitetura que permite a construção de ambientes compostos por uma sociedade com mais de um tutor e mais de um aluno;

Apesar dos avanços anteriormente descritos alguns pontos importantes no desenvolvimento deste projeto ainda podem vir a ser aperfeiçoados, entre os quais destacam-se:

- O aprimoramento de um modelo de coordenação das ações dos agentes apropriado para ser empregado por agentes pedagógicos em um ambiente inteligente de ensino;
- O aprimoramento do diagnóstico cognitivo e a forma de representação do modelo do aluno;
- O aprofundamento da especificação do fluxo de mensagens entre os agentes e de um modelo de máquina de estados para cada agente, modelando seu comportamento;
- O aprofundamento da especificação do modelo de estados mentais e a arquitetura interna dos agentes;
- O desenvolvimento de ferramentas de autoria mais adequadas que contemplem a modelagem do curso e a elaboração do material instrucional.
- O desenvolvimento de mecanismos de aprendizagem por parte dos agentes pedagógicos de tal forma que possam aprimorar o diagnóstico cognitivo e a escolha das estratégias de ensino mais adequadas.

Dois aspectos, no entanto, devem ser destacados como as principais propostas de avanço a partir deste projeto, as quais serão brevemente discutidas a seguir: a questão dos mecanismos de coordenação entre os agentes e a questão da modelagem do aluno.

Conforme foi explanado, o JADE adota o mecanismo de reconhecimento do processo executado pelos demais agentes, por meio do recebimento de mensagens que caracterizam determinados eventos. Entretanto, acredita-se que um mecanismo mais eficiente possa ser implementado futuramente dando mais poder ao ambiente como um todo e aumentando sua capacidade de adaptabilidade e de ajuda interativa

Diversos algoritmos de cooperação têm sido propostos para diferentes tipos de problemas. O algoritmo PGP (*Partial Global Planing*) apresentado por [DUR 91], posteriormente aprimorado por [DEC 92], [DEC95]e [LÊS 98]poderá vir a ser empregado neste trabalho. A metodologia empregada encontra respaldo na definição anteriormente discutida: cada agente de um sistema multiagentes possui a capacidade de obter, durante seu trabalho, informações a respeito do trabalho dos outros agentes. Cada agente pode utilizar este conhecimento para otimizar suas próprias tarefas.

A condição básica para o uso do algoritmo PGP é a situação na qual vários agentes distribuídos trabalham na solução de um problema geral. Um agente, como parte do PGP, observa as ações e os relacionamentos entre um grupo de agentes e estabelece as conclusões para o seu próprio trabalho. Este conhecimento é denominado *Partial Global Plan*, porque reflete o conhecimento parcial do plano determinado por um agente para ajudar a resolver um problema global. A figura 37, a seguir, mostra um exemplo que ilustra o princípio do PGP.

Dois agentes trabalham em dois subproblemas A e B que fazem parte de um problema global. Cada subproblema, por sua vez, consiste também em uma série de subproblemas (A1, A2, A3, B1 e B2). Cada agente passa informações sobre seu estado atual ao seu parceiro. No exemplo, o agente 1 informou ao agente 2 que está no momento trabalhando no subproblema A. De modo similar, o agente 2 comunicou ao agente 1 que está operando para solucionar o problema B.

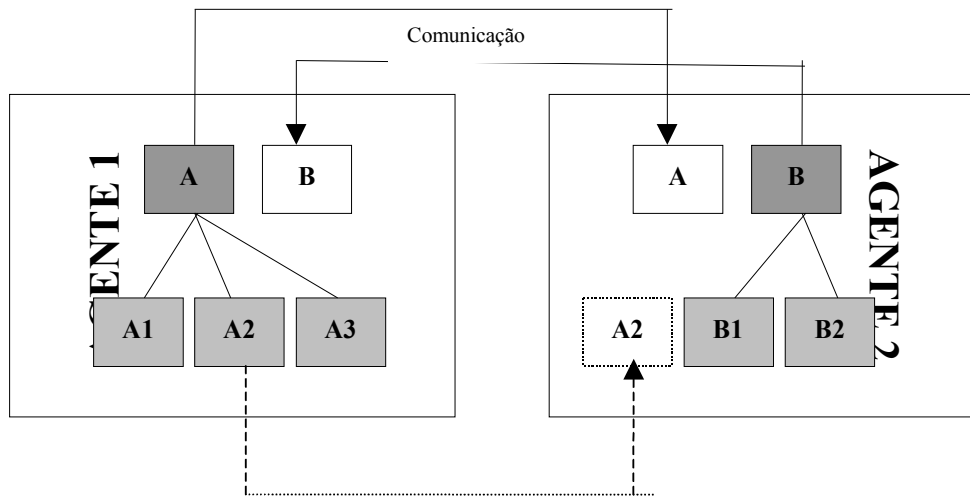


FIGURA 37 – Processo PGP

Cada um dos agentes utiliza esta informação, de acordo com a influência da situação em que seu parceiro se encontra em seu próprio trabalho. Por exemplo: se o agente 1 reconhece que seu subproblema A2 é dependente do subproblema B do agente 2, ele informa, de modo apropriado a este agente. Em alguns casos, isto pode ter como resultado, que o subproblema A2 seja transferido completamente para o agente 2, pois ele teria que processá-lo de qualquer forma como parte do subproblema B.

O processo PGP pode ser subdividido em quatro passos consecutivos:

1. **Criação dos planos locais de cada agente.** Embora esta etapa, a rigor, não faça parte do PGP propriamente dito, deve ser considerado que um agente tenha a capacidade de adaptar dinamicamente seus planos originais.
2. **Comunicação e intercâmbio de planos entre os agentes.** Cada agente deve possuir certa organização de modo a poder decidir que papel os outros agentes desempenham no processo de solução do problema e para quais agentes e que informações interessam, ou seja, um agente deve poder decidir que partes da sua informação ele deve passar a cada agente.
3. **Criação do PGP.** Uma vez que um agente tenha recebido informações sobre os planos de outros agentes, ele deve verificar se as novas informações contêm dependências em relação aos seus próprios planos e agrupar os subplanos em unidades lógicas resultando em um “Plano Global Parcial” (PGP), contendo os seguintes componentes:
 - **Objetivos;**
 - **Mapa do plano de atividades;**
 - **Grafo de construção da solução;**
 - **Status.**
4. **Modificação e otimização do PGP.** Um plano local modificado é produzido, otimizando o processo de busca de resultados.

As maiores vantagens do emprego de um algoritmo deste tipo são a obtenção de um comportamento altamente dinâmico do sistema, pois os planos são constantemente readaptados às novas situações ambientais, produzindo uma grande flexibilidade e eficácia. Além disso, evita-se quase por completo qualquer atividade redundante no sistema, aumentando sua eficiência.

As principais restrições ao uso deste algoritmo dizem respeito à dificuldade de implementá-lo em sistemas constituídos por agentes heterogêneos; a diminuição da sua eficácia em ambientes onde ocorrem mudanças repetitivas e em situações que demandem respostas em tempo real. Extensões deste modelo, no entanto, apresentadas por [DEC 92], [DEC 95] e [LES 98] procuram contornar estas limitações.

Com relação à questão da modelagem do aluno em ambientes de ensino baseados em WWW, [ZHU 99] resume a questão da seguinte forma:

Os sistemas de Educação a Distância baseados em WWW não possuem restrições de tempo nem de espaço. Os alunos podem interagir com o sistema em qualquer hora e em qualquer lugar. Além disso, as ferramentas disponíveis permitem a comunicação entre os alunos e professores muito facilmente, permitindo realimentação de forma quase imediata. Os alunos e professores podem compartilhar informações entre si e excelentes estratégias de ensino podem ser estabelecidas através dos recursos disponíveis na rede, em todo o mundo. Afora isso, é possível dispor de conhecimentos extensos e avançados jamais permitidos até então. Os estudantes podem decidir o que aprender, como aprender e quando aprender, privilegiando metodologias de ensino centradas no aluno e de fundo exploratório e construtivista.

Nem tudo, no entanto, são vantagens no ensino baseado em WWW. Alguns aspectos importantes devem ser considerados:

- A maioria dos sistemas de Educação a Distância baseados em WWW não possuem inteligência e adaptação.
- Os alunos costumam perder-se quando precisam navegar escolhendo caminhos entre os labirintos de *links* disponibilizados nas páginas da HTML.
- As páginas HTML, por si só, não constituem um sistema de ensino.
- É muito difícil para o aluno, por si só, obter material que seja de interesse para ele, no meio do universo de material disponível.

Conforme pode ser observado nos trabalhos correlatos apresentados no capítulo 2.4, as pesquisas neste sentido têm-se voltado para três grandes correntes: o uso de páginas WWW adaptativas, que utilizam algum método para modificar o conteúdo das páginas para adaptá-las às atitudes do aluno; o uso de sistemas WWW baseados em ITS, que utiliza a arquitetura tradicional de Tutores Inteligentes e usa uma interface WWW, incluindo, algumas vezes mecanismos de aprendizagem colaborativa; e as arquiteturas que utilizam agentes inteligentes, como no caso da arquitetura proposta pelo JADE.

Todas elas, no entanto têm em comum uma forte dependência de uma correta e robusta modelagem do aluno. É o modelo do aluno o principal instrumento pelo qual é possível prover táticas de ensino individualizadas que reflitam o nível de conhecimento individual de cada aluno, suas habilidades de aprendizagem e seus objetivos. O Modelo de aluno registra os erros do aluno de tal modo que o sistema possa prover estratégias de ensino adequadas para revisar os conteúdos. Assim, quanto mais preciso for, este modelo, maior e melhor é a adaptabilidade do sistema.

[ZHU 99] cita três tipos de modelo de aluno: o modelo de *Overlay Student Model* o *Buggy Model* e o *Cognitive Student Model* o detalhamento desses modelos pode ser encontrado em [SIL 96a].

A proposta desse autor para a modelagem de alunos em sistemas distribuídos de ensino inteligentes é a utilização de um modelo composto denominado *Compound Student Model* que combina o *Overlay Student Model* e o *Cognitive Student Model*

O *Overlay Student Model* é utilizado para representar o subconjunto do modelo do conhecimento do domínio que o aluno possui. Um modelo hierárquico é construído para representar o conhecimento dividindo-o em quatro níveis: curso, capítulo, seção e conceito e montado em uma estrutura de árvore que contempla os relacionamentos de pré-requisitos e co-requisitos das unidades. Nesta estrutura são assinalados os avanços percorridos pelo aluno na medida em que o sistema percebe que ele obteve sucesso em cada uma das unidades. Este modelo permite ao sistema saber o que o aluno aprendeu ou não e o que pode ser proposto a seguir, permitindo ao sistema guiar o aluno. Entretanto, este conjunto de informações ainda não é suficiente, pois com base nelas não é possível filtrar o material de ensino ou decidir que tática deve ser utilizada para aquele aluno em particular.

O *Cognitive Student Model* proposto por [ZHU 99] classifica o material que pode ser utilizado de acordo com quatro categorias de parâmetros: *habilidade cognitiva sintética*, *mídia de preferência* e *habilidade espacial*. O modelo do conhecimento é estendido para contemplar esta classificação, atribuindo um grau de 0 a 3 em cada um destes parâmetros para as várias alternativas existentes de material instrucional em uma mesma unidade e o modelo do aluno registra a alternativa mais adequada para o aluno, permitindo que o sistema possa filtrar o material a ser apresentado de acordo com esta classificação.

Este projeto tem um caráter nitidamente multidisciplinar e buscou aglutinar alguns avanços obtidos em diversas áreas envolvidas em Educação a Distância, buscando obter resultados práticos no campo da pesquisa experimental aplicada. Além disso, foi resultado da contribuição de vários trabalhos agregados, que já foram citados, sem os quais não seria possível chegar a este resultado.

Esperamos com este trabalho ter trazido alguma contribuição no sentido de aprimorar a eficácia dos ambientes computacionais voltados ao processo educacional, agregando conceitos de diferentes áreas através do estabelecimento de uma metodologia e da implementação de um sistema de DILE, utilizando uma arquitetura funcional baseada em uma sociedade de agentes. E, acima de tudo, esperamos que este trabalho tenha prosseguimento e continue servindo de base para muitos outros, como já tem sido até aqui.

Bibliografia

- [AND 97] ANDRE, E.; MULLER, J.; RIST, T. Life-Like Presentations Agents: A New perspective for Computer Based Technical Documentation. In: WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, AI-ED, 8., 1997. Proceedings... Kobe, Japan: [s.n.], 1997.
- [ARP 98] ARPA - Advanced Research and Projects Agency. **KSE - Knowledge Sharing Effort**. Disponível em <<http://www.cs.umbc.edu/kse/>>. Acesso em: 21 de jan.1998.
- [ASC 2000] ASCENCIO, Ana Fernanda Gomes. **Método heurístico para projeto de interfaces inteligentes com usabilidade**. Porto Alegre: CPGCC da UFRGS, 2000. Dissertação de Mestrado.
- [AUS 99] DMARS. AUSTRALIAN ARTIFICIAL INTELLIGENCE INSTITUTE. Disponível em: <<http://www.aaii.oz.au/proj/dMARS-prod-brief.html>>. Acesso em: 01 de mar.1999.
- [BAR 82] BARR, A. ; FEIGENBAUM, E. A. **The handbook of Artificial Intelligence**. Los Altos: Kaufman, 1982.
- [BAR 99] BARBUCEANU, M; FOX, M.S. **The Design of COOL: A Language for Representing Cooperation Knowledge in MultiAgent Systems**. Disponível em: < <http://www.ie.utoronto.ca/EIL/ABSpage/ABS-intro.html>>. Acesso em: 01 de mar.1999).
- [BAK 97] BAKER, Albert. **JAFMAS – A java-based agent framework for multiagent systems. Development and Implementation**. Cincinnati: Department of Electrical & Computer Engineering and Computer Science University of Cincinnati, 1997. Doctoral thesis.
- [BIC 98] BICA, Francine; SILVEIRA, Ricardo et al. Eletrotutor III - uma abordagem multiagentes. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 8., 1998. **Anais...** Fortaleza: [s.n.],1998
- [BIC 99] BICA, Francine. **Eletrotutor III: Uma abordagem multiagente para o ensino a distância**. Porto Alegre: CPGCC da UFRGS, 1999. Dissertação de Mestrado.
- [BOR 94] BORDINI, Rafael Heitor. **Suporte lingüístico para migração de agentes**. Porto Alegre: CPGCC da UFRGS,1994. Dissertação de Mestrado
- [BRA 97] BRADSHAW, J. M. An introduction to software agents In: BRADSHAW, J. M. Ed. **Software Agents**. Massachusetts: MIT Press, 1997.
- [BRE 98] BRENNER, W.; RÜDIGERr, Z.; WITTIG, H. **Intelligent Software Agents. Foundations and applications**. Berlin: Springer-Verlag, 1998.
- [CHE 95] CHEIKES, B. A. GIA: An Agent Based Architecture for Intelligent Tutoring Systems. In: THE CIKM WORKSHOP ON INTELLIGENT INFORMATION AGENTS, 1995. **Proceedings...** [S.l.: s.n],1995

- [CLA 90] CLANCEY, W. J. ; SOLOWAY, E. Artificial Intelligence and learning environments. **Artificial Intelligence**, Amsterdam, v.1, n.42, p.1 - 6, Feb. 1990.
- [COA 93] COAD, P.; YOURDON, E. **Projeto baseado em objetos**. Rio de Janeiro: Campus, 1993.
- [COO 97] COOK, J. An Empirically Based mentoring Agent for Supporting Music Composition Learning. In: WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, AI-ED, 8., 1997. **Proceedings...** Kobe, Japan: [s.n.], 1997.
- [COR 94] CORREA, M. **The architecture of dialogues among distributed cognitive agents**. Rio de Janeiro: Universidade Federal do Rio de Janeiro, Pós-Graduação em Engenharia, 1994. Tese de doutorado.
- [COS 92] COSTA, Ernesto (Ed.). **New directions for Intelligent Tutoring Systems**. Berlin: Springer-Verlag, 1992.
- [COS 97] COSTA, E. B. Perkusich, A. A Multi-Agent Interactive Learning Environment Model. In: WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, AI-ED, 8., 1997. **Proceedings...** Kobe, Japan: [s.n.], 1997.
- [COS 92] COSTA, Marcos Mota do Carmo. **Introdução à lógica modal aplicada à Computação**. Porto Alegre: Instituto de Informática da UFRGS, 1992. Trabalho apresentado na Escola de Computação, 8.,1982, Gramado.
- [COS 97] COSTA, E. B. Perkusich, A. A Multi-Agent Interactive Learning Environment Model. In: WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, AI-ED, 8., 1997. **Proceedings...** Kobe, Japan: [s.n.], 1997.
- [DAM98] D'AMICO, Carmen Barbosa de et al. Adapting teaching strategies in a learning environment on WWW. In: WORLD CONFERENCE OF THE WWW, INTERNET & INTRANET, 3., 1998, Orlando, US. **Proceedings...** Charlottesville: AACE, 1998.
- [DAR 93] THE DARPA KNOWLEDGE SHARING INITIATIVE EXTERNAL INTERFACES WORKING GROUP. **Specification of the KQML Agente-Communication Language**. Baltimore: University of Maryland, 1993.
- [DAR 99] DARTMOUTH UNIVERSITY **AgentTcl, Transportable Agents System**, Computer Science Dept., Dartmouth University. Disponível em: <<http://www.cs.dartmouth.edu/agenttcl.html>>. Acesso em: 01 mar.1999.
- [DEC 92] DECKER, K.; LESSER, V. Generalized Partial Global Planning, **International Journal on Intelligent Cooperative Information Systems**, [S.l.], v.2, n.1, p. 319-346, June 1992
- [DEC 95] DECKER, K.; LESSER, V. Designing a Family of Coordination Algorithms. In: INTERNATIONAL CONFERENCE ON MULTI-AGENT SYSTEMS,1., 1995. **Proceedings...** [S.l.: s.n], 1995

- [DEM 93] DEMAZEAU, Yves. Distributed Artificial Intelligence and multiagents systems. In: SIMPÓSIO BRASILEIRO DE INTELIGÊNCIA ARTIFICIAL, 10., 1993 Porto Alegre. **Anais...** Porto Alegre: SBC, 1993.
- [DEE 98] DEEPIKA, Chauchan; BAKER, Albert. JAFMAS: a multiagent application development system In: INTERNATIONAL CONFERENCE ON AUTONOMOUS AGENTS, 2., 1998. **Proceedings...** [S.l.: s.n], 1998. p 100-107.
- [DIL 97] DILLENBOURG, P. et al. The Design of MOO Agents: Implications from an Empirical CSCW Study. In: WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, AI-ED, 8., 1997. **Proceedings...** Kobe, Japan: [s.n.], 1997.
- [DOR 97] DORAN, J. E.; FRANKLIN, S.; JENNINGS, N. R. ; NORMAN, T. J. On cooperation in multiagent systems. **The Knowledge Engineering Review**, London, v. 3, n. 12, 1997. Disponível em: <<http://www.elec.qmw.ac.uk/dai/pubs/fomas.html>>. Acesso em: 20 out.2000.
- [DUR 91] DURFEE, E.H.; LESSER, V.R. Partial Global Planning: A Coordination Framework for Distributed Hypothesis Formation. **IEEE Transactions on Systems, Man, and Cybernetics**, New York, n. 21, v.5 p.1167-1183, September/October 1991.
- [ECH 99] ECHELON CORPORATION. **Echelon, About LonWorks Control Technology**. Disponível em: <<http://www.lonworks.echelon.com/>>. Acesso em: 01 mar.1999.
- [EDW 97] ENSINO a distância no WWW. [S.l.: s.n], Disponível em: <<http://www.penta.ufrgs.br/edu/eduwww.html>>. Acesso em: 10 jul.1997.
- [EYS 94] EYSENCK, M.; KEANE, M. **Psicologia Cognitiva: um manual introdutório**. Porto Alegre: Artes Médicas, 1994.
- [FIN 92] FININ, T et al. **Specification of the KQML Agent Communication Language**. Palo Alto: Enterprise Integration Technologies, 1992 (Technical report EIT TR 92-04). Disponível em <<http://www.cs.umbc.edu/kqml/kqmlspec/spec.htm>>. Acesso em: 01 jan.1998.
- [FIN 98] FININ, T **Ontology's for Knowledge Sharing**. disponível em <<http://www.cs.umbc.edu/kse/ontology/>>. Acesso em: 01 jan.1998.
- [FRA 97] FRANKLIN, S.; GRAESSER, A. Is it na agent or just a program? A taxonomy for autonomous agents, In: INTERNATIONAL WORKSHOP ON AGENT THEORIES, ARCHITECTURES AND LANGUAGES, 3., 1997. **Proceedings...** Berlin: Springer-Verlag, 1997.
- [FRE 77] FREIRE, Paulo. **Pedagogia do oprimido**. Rio de Janeiro: Paz e Terra, 1977.
- [FUK 95] FUKUHARA, Yoshimi et al. A Knowledge-based educational environment integrating conceptual knowledge and procedural knowledge in telecommunication service field. Tokyo: NTT, 1995. Disponível em: <<http://i-calat.min.ntt.jp>>. Acesso em: 12 nov.1997.
- [FUR 98] FURLAN, José Davi. **Modelagem de objetos através da UML**. São Paulo: Makron Books, 1998.

- [GEN 99] GENERAL MAGIC INC. **Odyssey Frequently Asked Questions**. Disponível em: <<http://www.genmagic.com/agents/odysseyfaq.html>>. Acesso em: 01 mar.1999.
- [GEN 95] GENESERETH, Michael R. **Knowledge Interchange Format Specification** draft for an American National Standard. 1995. Disponível em <<http://logic.stanford.edu/kif/specification.html>>. Acesso em 12 jan.1998.
- [GEN 99a] GENERAL MAGIC INC. **Telescript Technology: The Foundation for the Electronic Marketplace**. Disponível em: <<http://www.genmagic.com/Telescript/Whitepapers/wp1/whitepaper1.html>>. Acesso em: 01 mar.1999.
- [GER 99] GERMAN ARTIFICIAL INTELIGENCE RESEARCH INSTITUTE **InteRRap Agent Architecture**. Disponível em: <<http://www.dfki.uni-sb.de/~pischel/interrap.html>>. Acesso em: 01 mar.1999.
- [GIR 91] GIRAFFA, Lucia M. **Reflexões sobre o computador na escola**. Porto Alegre: PUCRS, 1991. Dissertação de Mestrado.
- [GIR 97] GIRAFFA, L. M.; NUNES, M. A.; VICCARI, R. M. Multi-Ecológico: Ambiente de ensino inteligente para suporte de educação ambiental. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 8., 1997. **Anais...** São José dos Campos: [s.n.], 1997.
- [GIR 98] GIRAFFA L. M. ; VICCARI R. M. **The use of Agents techniques on Intelligent Tutoring Systems**. In: CONGRESSO IBERO-AMERICANO DE INFORMÁTICA NA EDUCAÇÃO – RBIE,4., 1998. **Proceedings...** [S.l.: s.n],1998.
- [GIR 98a] GIRAFFA L. M., VICCARI R. M. ; SELF, J. **Multi-Agent based pedagogical games**. In: ITS, 4., 1998. **Proceedings...** [S.l.: s.n],1998.
- [HIE 97] HIETALA, P.; NIEMIREPO, T. Collaboration with Software Agents: What if the Learning Companion Agent Make Errors? In: WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, AI-ED, 8., 1997. **Proceedings...** Kobe, Japan: [s.n.], 1997.
- [HÜB 94] HÜBNER, Jomi Fred. **Migração de agentes em sistemas multi-agentes abertos**. Porto Alegre: CPGCC da UFRGS, 1994. Dissertação de Mestrado.
- [IBM 99] IBM TOKYO RESEARCH LABORATORIES. **IBM Aglets: Programming Mobile Agents in Java**. A White Paper. Disponível em: <<http://www.ibm.co.jp/trl/aglets/whitepaper.htm>>. Acesso em: 10 mar.1999.
- [JOH 97] JOHNSON, W. Lewis; SHAW, Erin. Using agents to overcome deficiencies in web-based courseware. In: WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, AI-ED, 8., 1997. **Proceedings...** [S.l.: s.n],1997. Disponível em: <<http://www.isi.edu/isd/johnson.html>>. Acesso em: 05 mar.2000.
- [KEE 91] KEEGAN, D. **Foundations of distance education**. 2nd ed. London: Routledge, 1991 apud NUNES, Ivônio Barros. **Noções de educação à**

distância. Disponível por WWW em <http://www.ibase.org.br/~ined/ivonio1.html> (out. 1997).

- [KIN 95] KINNY, D.; GEORGEFF, M. **A methodology and modeling technique for systems of BDI-Agents.** Melbourne: Australian AI Institute, 1995. Technical Report 58. Disponível em: <<http://www.aaii.oz.au>>. Acesso em: 15 mar.2000.
- [LAW 87] LAWER, R.; YAZDANI, M. **Artificial Intelligence and Education.** Norwood, New Jersey: Ablex Publishing, 1987. v.1.
- [LER 96] LEROUX, P. Co-operation between a Pedagogical Assistant, a Group of Learners and a Teacher. In: EUROPEAN CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, AI-ED, 1996. **Proceedings...**Lisbon: Colibri,1996.
- [LES 97] LESTER, J. et al. Mixed Initiative Problem Solving with Animated Pedagogical Agents. In: WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, AI-ED, 8., 1997. **Proceedings...** Kobe, Japan: [s.n.], 1997.
- [LES 98] LESSER, Victor. Reflections on the Nature of Multi-Agent Coordination and Its Implications for an Agent Architecture. In: **Autonomous Agents and Multi-Agent Systems.** [S.l.]: Kluwer Academic Publishers, 1998. v.1, p. 89-111.
- [LEW 97] LEWIS, W. J.; SHAW, E. Using Agents to Overcome Deficiencies in Web-Based Courseware. In: WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, AI-ED, 8., 1997. **Proceedings...** Kobe, Japan: [s.n.], 1997.
- [MAD 89] MADEIRA Milton. **Construção de Modelos de Desenvolvimento e da Formação de Conceitos em Crianças Brasileiras.** Porto Alegre: Pontifícia Universidade Católica do Rio Grande do Sul, 1989. Projeto de Pesquisa em Psicologia / FAPERGS / CNPq.
- [MAS 96] MATHOFF, J.;VAN HOE, R. APEALL: A Multi-agent approach to interactive learning environments. In: EUROPEAN WORKSHOP ON MODELING AUTONOMOUS AGENTS MAAMAW, 6., 1994. **Proceedings...** Berlin: Springer-Verlag, 1996.
- [MIZ 86] MIZUKAMI, Maria da Graça Nicoletti. **Ensino:** As abordagens do processo. São Paulo: EPU, 1986.
- [MIT 99] MITSUBISHI ELECTRIC INFORMATION TECHNOLOGY CENTER OF AMERICA. **Concordia, Mobile Agent Computing A White Paper.** [S.l.: s.n], 1999. Disponível em: <<http://www.meitca.com/HSL/Projects/Concordia/MobileAgentsWhitePaper.html>>. Acesso em: 15 mar.2000.
- [MOR 97] MORIN, J. F.; LELOUCHE, R. Tutoring Knowledge Modelling as Pedagogical Agents in an ITS. In: WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, AI-ED, 8., 1997. **Proceedings...** Kobe, Japan: [s.n.], 1997.
- [MOR 97a] MÓRA, Michael; LOPES, J. G.; VICCARI, R. Modelling dynamic aspects of intentions. In: PORTUGUESE CONFERENCE ON

- ARTIFICIAL INTELLIGENCE, 8., 1997. **Proceedings...** Berlin: Springer-Verlag, 1997.
- [MOR 98] MÓRA, Michael C. et al. BDI Models and Systems: Reducing the Gap. In: WORKSHOP ON AGENT THEORIES, ARCHITECTURES, AND LANGUAGES, 5., 1998. **Proceedings...** Berlin: Springer-Verlag, 1998.
- [MÜL 96] MÜLLER, J. P. **The design of intelligent agents**: a layered approach. Heidelberg: Springer-Verlag, 1996. (Lecture Notes in Computer Science, v. 1177).
- [MUR 97] MURRAY, W. R. Knowledge-based Guidance in the CAETI Center Associate. In: WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, AI-ED, 8., 1997. **Proceedings...** Kobe, Japan: [s.n.], 1997.
- [NAK 96] NAKABAYASHI, Kiyoshi et al. **An intelligent Tutoring system on word-wide-web**: towards an integrated learning environment on a distributed hypermedia. Tokyo: NTT, 1996. Disponível em <<http://i-calat.min.ntt.jp>>. Acesso em: 04 mar.997.
- [NAK 97] NAKABAYASHI, Kiyoshi et al. Architecture of na intelligent tutoring system on the WWW. In: WORLD CONFERENCE OF THE AIED SOCIETY, 8., 1997, Kobe. **Proceedings...** [S.l.: s.n.], 1997.
- [NOR 2000] NORMAN, Timothy J.; JENNINGS, Nicholas R. Constructing a virtual training laboratory using intelligent agents. **International Journal of Continuous Engineering and Life-long Learning**, [S.l.], 2000.
- [NTT 99] NTT COMMUNICATION SCIENCE LABORATORIES; ISHIDA LABORATORY **AgentTalk**: Describing Multiagent Coordination Protocols. Disponível em: <<http://www.cslab.tas.ntt.jp/at/>>. Acesso em: 12 mar.1999.
- [NUN 97] NUNES, Ivônio Barros. **Noções de educação à distância**. Disponível em: <<http://www.ibase.org.br/~ined/ivonio1.html>>. Acesso em: 15 out.1997.
- [OII 99] OBJECT INTERNATIONAL INCORPORATION. **Página Institucional**. Disponível em: <<http://www.oi.com/>>. Acesso em: 15 mar. 1999.
- [OBJ 99] OBJECT SPACE INC. **Voyager Technical Review**. Disponível em: <http://www.objectspace.com/voyager/voyager_white_papers.html>. Acesso em: 15 mar.1999.
- [OLI 96] OLIVEIRA, F. M. ; VICCARI, R. M. Are learning systems distributed or social systems? In: EUROPEAN CONFERENCE ON A. I. IN EDUCATION. **Proceedings...** Lisbon: [s.n.], 1996.
- [PER 98] PEREIRA, Adriana Soares; D'AMICO, Carmen Barbosa de; VICCARI, Rosa Maria. Adaptative teaching and learning environment for advanced web based educational applications. In: IBEROAMERICAN WORKSHOP ON DISTRIBUTED ARTIFICIAL INTELLIGENCE AND MULTI-AGENT SYSTEMS, 2., 1998, Toledo. **Proceedings...** Madrid: CICYT, DIT-UPM, 1998. p. 161-165.

- [PER 98a] PEREIRA, Adriana Soares; D'AMICO, Carmen Barbosa de, GEYER, Cláudio Fernando Resin Uma aplicação de ensino orientada a agentes. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 8., 1998. **Anais...** Fortaleza: [s.n.],1998.
- [PET 98] PETERS, Otto, **Learning and teaching in Distance Education**. London: Kogan Page, 1998.
- [PIA 77] PIAGET, J. **A Psicologia**. Lisboa: Bertrand, 1977
- [POS 81] POSTEL, J. **Internet Protocol**. [S.l.]: The Internet Engineering Task Force (IETF), 1981. (RFC 0791). Disponível em: <<http://www.ietf.org>>. Acesso em: 05 jan.1998.
- [POS 81a] POSTEL, J. **Transmission Control Protocol**. [S.l.]: The Internet Engineering Task Force (IETF), 1981. (RFC 0793). Disponível em: <<http://www.ietf.org>>. Acesso em: 05 jan.1998.
- [RAM85] RAMIREZ, Maria del Sagrario. **Métodos de educação de adultos**. São Paulo: Loyola, 1985.
- [RAO 95] RAO, A.; GEORGEFF, M. BDI agents: From theory to practice. In: INTERNATIONAL CONFERENCE ON MULTIAGENT SYSTEMS ICMAS, 1., 1995. **Proceedings...** San Francisco: [s.n.], 1995.
- [RAO 96] RAO, A.; GEORGEFF, M. Modeling and design of multi-agent systems. In: ECAI WORKSHOP (ATAL), 1996. Budapest, Hungary, 1996. **Proceedings...**[S.l.]: Springer-Verlag, 1996.
- [REI 2000] REIS, Alessandro Boeira dos. **Um modelo do aluno adaptativo para sistemas na Web**. Porto Alegre: CPGCC da UFRGS, 2000. Dissertação de Mestrado.
- [RET 99] RETICULAR SYSTEMS INC. **Agent Builder White Paper**. Disponível em <<http://www.agentbuilder.com/Documentation/WhitePaper/>>. Acesso em: mar.1999.
- [RIC 97] RICKELL, J.; JOHNSON, W. L. Intelligent Tutoring in Virtual Reality: A Preliminary report. In: WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, AI-ED, 8., 1997. **Proceedings...** Kobe, Japan: [s.n.], 1997.
- [RIT 97] RITTER, S. Communication, Co-operation and Competition Among Multiple Tutor Agents. In: WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, AI-ED, 8., 1997. **Proceedings...** Kobe, Japan: [s.n.], 1997.
- [ROG 77] ROGERS, CARL RANSOM. **Liberdade para aprender**. Belo Horizonte: Interlivros, 1977.
- [SAN 99] THE SANTA FE INSTITUTE. **Swarm, The Swarm Simulation System**, Disponível em: <<http://www.santafe.edu/projects/swarm/overview.html>>. Acesso em: 05 mar.1999.
- [SEL 92] SELF, John A. Computational Mathetics: the missing link in intelligent tutoring systems research? In: COSTA, E.(Ed.). **New directions in intelligent tutoring systems research**. Berlin: Springer-Verlag, 1992.

- [SHE 96] SHERRY, L. Issues in Distance Learning. **International Journal of Distance Education**, v.1, n.4, p. 337-365, 1996 Disponível em: <<http://www.cudenver.edu/public/education/sherry/pubs/issues.html>>. Acesso em: 05 dez. 1997.
- [SHO 91] SHOHAM, Y. AGENT-0: A simple agent language and its interpreter. In: CONFERENCE ON ARTIFICIAL INTELLIGENCE, 9., 1991, Anaheim **Proceedings...** CA: MIT Press, 1991. p. 704-709
- [SHO 93] SHOHAM, Y Agent-oriented programming. **Artificial Intelligence**, Amsterdam, n.60, v.1, p.51 - 92, Feb. 1993.
- [SIL 92] SILVEIRA, Ricardo A. **Inteligência Artificial em Educação: um modelo de sistema tutorial inteligente para microcomputadores** Porto Alegre: PUCRS,1992. Dissertação de Mestrado em Educação.
- [SIL 94] SILVEIRA, Ricardo A. Sistemas tutoriais inteligentes: Inteligência Artificial em Educação. **Educação**, Porto Alegre, v. 17, n.27, p.127 - 135, 1994.
- [SIL 96] SILVEIRA, Ricardo A. **Um ambiente inteligente de ensino e aprendizagem (ITLE) estruturado em uma sociedade de agentes**. Porto Alegre: CPGCC da UFRGS, 1996. Projeto de trabalho de doutorado.
- [SIL 96a] SILVEIRA, Ricardo A. **Diagnóstico e modelagem cognitiva em ambientes de ensino inteligentes: trabalho individual**. Porto Alegre: CPGCC da UFRGS, 1996.
- [SIL 96b] SILVEIRA, Ricardo. **Eletrotutor II: um tutor na web: trabalho individual**. Porto Alegre: CPGCC da UFRGS, 1996.
- [SIL 97] SILVEIRA, R. A.; VICCARI, R. M. Projeto Eletrotutor: Desenvolvimento e Avaliação de Ambientes Inteligentes de Ensino-Aprendizagem. In: CONFERENCIA LATINO-AMERICANA DE INFORMATICA, 23., 1997. **Proceedings...** Valparaíso: CLEI, 1997.
- [SIL 97a] SILVEIRA, Ricardo et al. Desenvolvimento e avaliação de duas abordagens de ambientes de ensino inteligentes. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 8., 1997. **Anais...** São José dos Campos: [s.n.], 1997.
- [SIL 98] SILVEIRA, R. A. **Ambientes inteligentes distribuídos de aprendizagem**. Porto Alegre: CPGCC UFRGS, 1997. (EQ - 19).
- [SIM 81] SIMON, H. **As Ciências do artificial**. Coimbra: A. Amado, 1981.
- [SPO 95] SPODICK, Edward F. **The evolution of distance learning**. Hong Kong: University of Science & Technology Library, 1995. Disponível em: <<http://sqzm14.ust.hk/distance>>. Acesso em: 08 ago. 1995.
- [STA 99] STANFORD UNIVERSITY. **JATLite overview**. Disponível em: <http://java.stanford.edu/java_agent/htmlJATLiteOverview.html/>. Acesso em: 05 mar.1999.
- [SUN 98] SUN MICROSYSTEMS. **Java Remote Method Invocation – distributed computing for Java**. Disponível em: <<http://java.sun.com/marketing/collateral/javarmi.html>>. Acesso em: 05 jan.1998.

- [TED 97] TEDESCO, P.A.; BARROS, F. A.; SOUZA, F. SEI - Sistema de Ensino Inteligente In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 8., 1997. **Anais...** São José dos Campos: [s. n.], 1997
- [THO 94] Thomas, S. R. **The PLACA Agent Programming Language**. Berlin: Springer-Verlag, 1994.
- [TOR 95] TORSUN, I.S. **Foundations of Intelligent Knowledge-based systems**. London: Academic Press, 1995.
- [VIC89] VICCARI, R. M. **Um tutor inteligente para a programação em lógica: idealização, projeto e desenvolvimento**. Coimbra: Universidade de Coimbra, 1989. Tese de doutorado.
- [WAN 97] WANG, W. C.; CHAN, T. M. Experience of Designing an Agent-Oriented Programming Language for Developing Social Learning Systems. In: WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, 8., 1997. **Proceedings...** Tokyo: IOS Press, 1997.
- [WAI 95] WAINER, Ricardo; MADEIRA, Milton. Psicoterapias Cognitivas: convergências e divergências entre as abordagens Construtivistas e do Processamento da Informação. In: JORNADA DE TERAPIA COGNITIVA, 2., 1995. **Anais...** São Paulo: [s. n.], 1995.
- [WAV 92] WAVISH, Peter. Exploiting emergent behavior in Multi-agent systems In: WERNER, Eric ; DEMAZEAU, Yves (Eds.). **Decentralized A. I.** [S.l.]: Elsevier Science Publishers, 1992.
- [WEB 97] WEBER, Gerhard; SPECHT, Marcus. User modeling and adaptative navigation support in WWW – based tutoring systems. In: USER MODELING INTERNATIONAL CONFERENCE, 6., 1997. **Proceedings...** Viena: Springer-Verlag, 1997.
- [WOO 99] WOOLDRIDGE, M.; JENNINGS, N. R.; KINNY, D. A methodology for agent-oriented analysis and design. In: INTERNATIONAL CONFERENCE ON AUTONOMOUS AGENTS, 3., 1999. **Proceedings...** Disponível em: <<http://www.elect.qmw.ac.uk/dai/pub>>. Acesso em: 25 mar.1999.
- [ZHU 99] ZHURONG, Zhou; XIAOZHEN, Zhang. A compound student model in web –based ITS. In: INTERNATIONAL CONFERENCE ON COMPUTERS IN EDUCATION, ICCE, 7, 1999. **Proceedings...** Tokyo: IOS Press, 1999.